Function Computation over Networks: Efficient Information Processing for Cache and Sensor Applications

THÈSE N^{O} 6787 (2015)

PRÉSENTÉE LE 20 NOVEMBRE 2015

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE D'INFORMATION DANS LES SYSTÈMES EN RÉSEAUX
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Chien-Yi WANG

acceptée sur proposition du jury:

Prof. B. Rimoldi, président du jury
Prof. M. C. Gastpar, directeur de thèse
Prof. G. Kramer, rapporteur
Prof. Y.-H. Kim, rapporteur
Prof. E. Telatar, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Abstract

This thesis looks at efficient information processing for two network applications: content delivery with caching and collecting summary statistics in wireless sensor networks. Both applications are studied under the same paradigm: function computation over networks, where distributed source nodes cooperatively communicate some functions of individual observations to one or multiple destinations. One approach that always works is to convey all observations and then let the destinations compute the desired functions by themselves. However, if the available communication resources are limited, then revealing less unwanted information becomes critical. Centered on this goal, this thesis develops new coding schemes using information-theoretic tools.

The first part of this thesis focuses on content delivery with caching. Caching is a technique that facilitates reallocation of communication resources in order to avoid network congestion during peak-traffic times. An information-theoretic model, termed sequential coding for computing, is proposed to analyze the potential gains offered by the caching technique. For the single-user case, the proposed framework succeeds in verifying the optimality of some simple caching strategies and in providing guidance towards optimal caching strategies. For the two-user case, five representative subproblems are considered, which draw connections with classic source coding problems including the Gray–Wyner system, successive refinement, and the Kaspi/Heegard–Berger problem. Afterwards, the problem of distributed computing with successive refinement is considered. It is shown that if full data recovery is required in the second stage of successive refinement, then any information acquired in the first stage will be useful later in the second stage.

The second part of this thesis looks at the collection of summary statistics in wireless sensor networks. Summary statistics include arithmetic mean, median, standard deviation, etc, and they belong to the class of symmetric functions. This thesis develops arithmetic computation coding in order to efficiently perform in-network computation for weighted arithmetic sums and symmetric functions. The developed arithmetic computation coding increases the achievable computation rate from $\Theta((\log L)/L)$ to $\Theta(1/\log L)$, where L is the number of sensors. Finally, this thesis demonstrates that interaction among sensors is beneficial for computation of type-threshold functions, e.g., the maximum and the indicator function, and that a non-vanishing computation rate is achievable.

Keywords: Coded caching, content delivery networks, distributed computing, Gaussian multiple access channel, information redundancy, interactive computation, joint source–channel coding, multi-terminal source coding, wireless sensor networks.

Résumé

Cette thèse se penche sur le traitement efficace de l'information pour deux applications de réseau : la livraison de contenu avec mise en cache et la collecte des statistiques sommaires dans les réseaux de capteurs sans fil. Les deux applications sont étudiées selon le même paradigme : le calcul de fonctions dans les réseaux, où les nœuds de source distribués transmettent de manière coopérative des fonctions d'observations individuelles vers une ou plusieurs destinations. Une approche qui fonctionne toujours est de transmettre toutes les observations et d'ensuite laisser les destinations calculer les fonctions désirées de manière autonome. Toutefois, si les ressources de communication disponibles sont limitées, il devient alors critique de révéler moins d'informations non désirées. Centrée sur cet objectif, cette thèse développe de nouvelles méthodes de codage faisant recours à des outils de la théorie de l'information.

La première partie de cette thèse se concentre sur la livraison de contenu avec mise en cache. La mise en cache est une technique qui facilite la réaffectation des ressources de communication afin d'éviter la congestion du réseau pendant les périodes de pointe du trafic. Un modèle de la théorie de l'information, appelé codage séquentiel pour le calcul de fonctions, est proposé pour analyser les gains potentiels offerts par la technique de mise en cache. Pour le cas d'un utilisateur unique, le cadre théorique proposé permet de prouver l'optimalité de certaines stratégies simples de mise en cache et de donner des pistes pour l'élaboration de stratégies optimales de mise en cache. Pour le cas de deux utilisateurs, cinq sous-problèmes représentatifs sont pris en compte, qui font le lien avec des problèmes classiques de codage de source dont le système de Gray–Wyner, le raffinement successif et le problème de Kaspi/Heegard–Berger. Ensuite, nous abordons le problème de calcul distribué avec raffinement successif. Il est démontré que si la reconstruction complète de données est requise dans la deuxième étape du raffinement successif, alors toutes les informations recueillies au cours de la première étape seront utiles pour la deuxième étape.

La deuxième partie de cette thèse examine la collecte des statistiques sommaires dans les réseaux de capteurs sans fil. Les statistiques sommaires comprennent la moyenne arithmétique, la médiane, l'écart-type, etc., et ils appartiennent à la classe des fonctions symétriques. Cette thèse développe le codage pour calcul arithmétique afin d'effectuer efficacement le calcul en réseau de sommes arithmétiques pondérées et de fonctions symétriques. Le codage pour calcul arithmétique ainsi développé augmente le taux de calcul réalisable d'une valeur $\Theta((\log L)/L)$ à $\Theta(1/\log L)$, où L est le nombre de capteurs. Finalement, cette thèse démontre que l'interaction entre les capteurs est bénéfique pour le calcul de fonctions de seuil appliquées à des

histogrammes, comme par exemple la valeur maximale et la fonction indicatrice, et qu'un taux de calcul non nul est réalisable.

Mots-clés : mise en cache codée, réseaux de diffusion de contenus, calcul distribué, canaux gaussiens à accès multiple, redondance de l'information, calcul interactif, codage conjoint de source et de canal, codage de source pour terminaux multiples, réseaux de capteurs sans fil.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Michael Gastpar, for his constant support and guidance throughout my graduate studies. Over the past four years, Michael has always been nice and patient in hearing my vague thoughts and has given me valuable feedback. The current progress in caching could not have been reached without Michael's encouragement when I felt like giving up. Furthermore, I appreciate Michael's assistance in improving my writing skills.

It is a great pleasure and honor to have Bixio Rimoldi, Emre Telatar, Gerhard Kramer, and Young-Han Kim on my thesis committee. I am grateful to them for reviewing the draft of this thesis and attending my oral exam in person.

I was very fortunate to have Sang-Woon Jeon and Sung Hoon Lim as research collaborators in different stages of my doctoral journey. I would like to thank them for sharing their research experiences and interesting stories within the IT society. I am also grateful to Abbas El Gamal and Young-Han Kim for their excellent textbook "Network Information Theory," which has always been a valuable resource for knowledge and references.

My life in the LINX group has been quite enjoyable. Thanks to Chen Feng, Naveen Goela, Sang-Woon Jeon, Sung Hoon Lim, Giel Op 't Veld, Adriano Pastore, Saeid Sahraei, Jiening Zhan, and Jingge Zhu for all the discussions and memories. Special thanks to France Faille for her administrative support and the organization of various group activities. I am grateful to France and Adriano for the French translation of the abstract in my thesis.

Finally, I would like to thank my family for their support in whatever decisions I have made.

Contents

List of Figures

1 Introduction

Traditionally, redundancy refers to the difference between the number of bits to transmit a message and the number of bits of actual information in the message. In his seminal paper [1], Shannon established the foundation of lossless data compression. For a discrete memoryless source (DMS), say $\langle X \rangle$, Shannon showed that the rate of actual information in $\langle X \rangle$ is its entropy $H(X)$.[1] Then, we have a reference point and the redundancy in terms of rate can be defined formally as the difference between the compression rate R and the entropy $H(X)$, i.e.,

$$\Delta := R - H(X).$$

There exist many coding schemes, e.g., the Huffman coding, that can make the redundancy Δ arbitrarily close to zero as the length of codes increases.

Next, consider the case where two source sequences x_1^k and x_2^k, generated by a DMS $\langle X_1, X_2 \rangle$, are observed at two distributed nodes. If we naively compress the two sequences separately, the required sum rate is $H(X_1) + H(X_2)$. However, as shown by Slepian and Wolf in [2], we can achieve the sum rate $H(X_1, X_2)$ without any communication between Nodes 1 and 2. In this case, the redundancy can be defined as the difference between the sum of the individual compression rates R_1, R_2 and the joint entropy $H(X_1, X_2)$, i.e.,

$$\Delta_2 := R_1 + R_2 - H(X_1, X_2).$$

At this point, the vague term "actual information" can be interpreted either as the information that the destination *wants* or the information that the destination *does now know*. Then, one natural question comes to mind: Is it always possible to convey only the desired information, without revealing any unwanted information?

The answer turns out to be no. Namely, if we use the entropy of the desired information as the reference point, then redundancy is inevitable in general. An example is provided by Körner and Marton in [3] (see Section 2.6 for the details), where we need twice the entropy of the desired information. Another example is the problem of source coding with a helper. We will show in Section 2.2 that if the

[1] The definition of DMS can be found in the beginning of Chapter 2.

helper really participates in the communication, then it is inevitable that part of the information conveyed by the helper is unwanted. Therefore, although we can always avoid redundancy in representation, it is in general impossible to avoid *redundancy in information*. Since information redundancy is inevitable, the best we can do is to reduce it or even to exploit it.

In this thesis, we study how to deal with information redundancy for two network applications: content delivery with caching and collecting summary statistics in wireless sensor networks, under the paradigm of function computation over networks.

- **Content Delivery Networks with Caching**. In a small-scale content delivery network, there is only one server, who has access to a database and serves content to end users. A typical application is video streaming: End users first send their individual requests to the server. Then, the server fetches the videos from the database and delivers them to the end users. Note that some of the desired videos can be correlated or exactly the same. Usually in the evening, the server receives more requests than in the early morning and thus network congestion occurs more often. It is fair to say that the communication resources are evenly distributed over time. Thus, it is desirable to be capable of moving some workload from the evening period to the morning period.

  Recently, *caching* has drawn a lot of attention due to its high potential in reallocating the available communication resources. Before knowing the end users' requests, cleverly coded partial content is delivered to the end users and stored in private and/or shared caches. Since the requests are unknown, the usefulness of the cache content can not be guaranteed. It is likely that the cache content turns out to be redundant. Nevertheless, by carefully designing the caching strategy, we can increase its probability of being useful in the delivery stage. Therefore, caching is a technique for moving communication resources from the low-traffic time to the peak-traffic time by tolerating some information redundancy.

- **Wireless Sensor Networks**. In a wireless sensor network, multiple spatially distributed autonomous sensors monitor physical and environmental conditions and report their observations to the fusion center, which analyzes the collected sensor data and takes necessary actions. Typical applications of wireless sensor networks include air/water quality monitoring and forest fire detection. Sensor deployment can be costly, so the lifetime of sensors should be months or even years. Therefore, *power efficiency* becomes an important issue for system design.

  Traditionally, sensors simply convey all the measured parameters to the fusion center. However, for many applications, the fusion center is only interested in acquiring an *indication* or, more generally, a *function* of the parameters, rather than the parameters themselves. Some common functions include average, median, standard deviation, etc, which are summary statistics. Another example is forest fire detection for which only an alarm signal is needed instead of the whole temperature and/or humidity readings. If the desired function is not injective (one-to-one), then conveying all the measured parameters reveals redundant information which consumes valuable communication

resources. Therefore, if possible, it is important to develop coding schemes that can reveal less redundant information.

1.1 Contributions

- **Sequential Coding for Computing** (Chapters 3 and 4): Content delivery with caching is formulated as a multi-terminal source coding problem with side information. The proposed framework naturally takes care of any interrelation among the various requested contents. Besides, many coding techniques and insights can be directly borrowed from the well-developed source coding literature. In particular, we prove rigorously that some intuitive caching strategies are indeed optimal for the single-user case. Additionally, when the contents are requested equally likely, we show that the usefulness of the cache content can be measured by conditional total correlation, which is Wyner's common information for the case of two components. From a theoretical point of view, this new class of problems draws interesting connections with many classic source coding problems. Two principles resulting from sequential coding and the Gray–Wyner system are presented. These two principles assist in identifying manageable subproblems in the general multi-user case, for each of which a single-letter characterization of the optimal rate region is attainable.

- **Distributed Computing with Successive Refinement** (Chapter 5): The classic successive refinement problem is extended to the distributed setting with an emphasis on function computation. We are interested in *successive refinability*, which refers to a property of sources and the desired functions that successively computing the two functions is without loss of optimality. We restrict attention to the special case where the source sequences have to be recovered losslessly in the second stage. Both source coding and joint source–channel coding are considered. In source coding, we show that all sources are successively refinable in sum rate, no matter which function has to be recovered in the first stage. In joint source–channel coding, the sources are assumed independent. For a class of multiple access channels (MAC), we show that all sources are successively refinable with respect to a class of linear functions. Finally, when the sources have equal entropy, we provide a simple sufficient condition of successive refinability for partially invertible functions.

- **Function Computation over Linear Multiple Access Channels** (Chapters 6 and 7): This part of work is an extension of linear computation coding developed by Nazer and Gastpar [4]. Based on the observation that a modulo sum remains an arithmetic sum as long as there is no "wrap around," we develop *arithmetic computation coding* for computing weighted arithmetic sums, weighted modulo sums, frequency histogram, and general symmetric functions over the Gaussian MAC. The arithmetic computation coding achieves the worst-case computation rate $\Theta\left(\frac{1}{\log L}\right)$, where L is the number of sensors. Note that conveying the full data only achieves a computation rate scaling as $\Theta\left(\frac{\log L}{L}\right)$ in the worst case. Assuming noiseless causal feedback is available, we show that interaction among sensors, enabled by the feedback link, assists

in providing a non-vanishing computation rate for the class of type-threshold functions, which is a subclass of symmetric functions.

1.2 Outline

In Chapter 2, we begin with a review of some classic source coding problems in network information theory including Slepian–Wolf coding, source coding with a helper, the Gray–Wyner system, and the Kaspi/Heegard–Berger problem. Then, we review several developments for function computation which includes lossless coding for computing, the Körner–Marton problem, and linear computation coding.

Chapters 3 and 4 are devoted to the problem of sequential coding for computing for the single-user case and the two-user case, respectively. We start with a discussion on the modeling of content delivery networks. For the single-user case, we present a single-letter characterization of the optimal rate region and discuss the following three cases in details: independent components, nested components, and uniform request. Then, we discuss the distributed compress–bin scheme with successive decoding and the effect of changing decoding order. We end Chapter 3 with a look at the single-request model. In Chapter 4, five representative subproblems of the two-user case are considered. The first two subproblems are extensions of the Gray–Wyner system and the third subproblem is a special case of distributed computing with successive refinement. The last configuration draws connection with the Kaspi/Heegard–Berger problem. Finally, we provide a general achievability for the entire system, which includes the achievable schemes of the five subproblems as special cases.

Chapter 5 looks at the problem of distributed computing with successive refinement. We consider both source coding and joint source–channel coding. The emphasis is placed on the case where the source sequences have to be recovered losslessly in the second stage.

Chapter 6 develops arithmetic computation coding for linear MACs. We start with computing an arithmetic sum over a modulo-3 MAC to gain some insight. Next, we demonstrate how to compute an arithmetic sum over a Gaussian MAC efficiently and then extend to frequency histogram and general symmetric functions. We end with a look at function computation over a symmetric Rayleigh fading MAC.

We consider the symmetric Gaussian MAC with noiseless causal feedback in Chapter 7. The emphasis is placed on type-threshold functions. We first introduce a set of auxiliary random variables, also termed *descriptions*, with an analysis on its entropy. These descriptions serve as building blocks for interactive coding for computing. Building upon the arithmetic computation coding in Chapter 6 and the introduced descriptions, the proposed multi-round group broadcast is presented.

Finally, a conclusion is drawn in Chapter 8.

1.3 Notation and Terminology

Denote by $(\mathbb{R}, +, \times)$ the field of real numbers and by $(\mathbb{F}_q, \oplus_q, \otimes_q)$ the finite field of order q, where q is assumed to be prime throughout the thesis. Sometimes we drop the subscript if $q = 2$. We denote by $\mathbb{Z}^+$ the set of positive integers and $\mathbb{N} := \mathbb{Z}^+ \cup \{0\}$. Let $\sum$ denote the summation over $\mathbb{R}$ and $\bigoplus$ denote the summation

over a finite field, whose order will be clear from context. Throughout the thesis, all logarithms are to base two.

We use calligraphic symbols (e.g., $\mathcal{S}$) to denote sets. However, the symbols $\mathcal{E}$, $\mathcal{D}$, and $\mathcal{R}$ are preserved to denote encoding functions, decoding functions, and rate regions, respectively. Denote by $|\cdot|$ the cardinality of a set. We denote $\mathcal{A}\backslash\mathcal{B} := \{x \in \mathcal{A} | x \notin \mathcal{B}\}$. Random variables and their realizations are represented by uppercase letters (e.g., S) and lowercase letters (e.g., s), respectively. The probability of an event $\mathcal{A}$ is denoted by $\mathbb{P}(\mathcal{A})$ and the expectation (or expected value) of a random variable X is denoted by $\mathbb{E}[X]$. The probability distribution of a random variable X is denoted by p_X. We say that $X \multimap Y \multimap Z$ form a Markov chain if $p_{X,Y,Z} = p_Y p_{X|Y} p_{Z|Y}$. $X = \emptyset$ means that X is degenerated, i.e., $p_X(x) = 1$ for some arbitrary $x \in \mathcal{X}$. Denote by $\mathbb{1}\{\cdot\}$ the indicator function of an event and by $\mathbf{1}$ an all-one vector with appropriate dimension.

We denote $x^+ := \max\{x, 0\}$ for all $x \in \mathbb{R}$, $\log^+(x) := \max\{\log(x), 0\}$ for all $x \geq 0$, and $[x] := \{1, 2, \cdots \lceil x \rceil\}$ for all $x \geq 1$. Given any sequence, tuple, or vector $(x_1, x_2, \cdots, x_k)$, we use two short-hand notations $x^{\mathcal{J}}$ and $x_{\mathcal{J}}$ for the subsequence $(x_i : i \in \mathcal{J})$, for all $\mathcal{J} \subset [k]$. For the case $\mathcal{J} = [k]$, we use the notation x^k, $x_{[k]}$, and $\mathbf{x}^T$ interchangeably, where $\mathbf{x}^T$ is the transpose of the vector $\mathbf{x}$. With an abuse of notation, we also use the notations $\{x_i\}$ and $\{x_i\}_{i\in\mathcal{J}}$ to denote a sequence $(x_i : i \in \mathbb{Z}^+)$ and a tuple $(x_i : i \in \mathcal{J})$, respectively. We use boldface symbols (e.g., $\mathbf{H}$) to denote matrices. Denote by $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^k x_i^2}$ the Euclidean norm of the vector $\mathbf{x} \in \mathbb{R}^k$. The binary entropy function $h_\mathsf{b} : [0, 1] \to [0, 1]$ is defined as

$$h_\mathsf{b}(q) = \begin{cases} -q\log(q) - (1-q)\log(1-q) & \text{if } q \in (0, 1), \\ 0 & \text{if } q \in \{0, 1\}. \end{cases}$$

The usual notation for entropy, $H(X)$, and mutual information, $I(X;Y)$, is used. We follow the ϵ–δ notation in [5]. We follow the robust typicality introduced in [6]. For $X \sim p_X$ and $\epsilon \in (0, 1)$, the set of typical sequences of length k with respect to the probability distribution p_X and the parameter ϵ is denoted by $\mathcal{T}_\epsilon^{(k)}(X)$, which is defined as

$$\mathcal{T}_\epsilon^{(k)}(X) := \left\{ x^k \in \mathcal{X}^k : \left| \frac{\#(a|x^k)}{k} - p_X(a) \right| \leq \epsilon p_X(a), \forall a \in \mathcal{X} \right\},$$

where $\#(a|x^k)$ is the number of occurrences of a in x^k.

Given two functions $f(n)$ and $g(n)$, we say that $f(n) = O(g(n))$ if there exists $k > 0$ and n_0 such that for all $n > n_0$, $f(n) \leq kg(n)$. We say that $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$. Finally, we say that $f(n) = \Theta(g(n))$ if it holds that $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

2 Preliminaries

This chapter consists of two parts. In the first part, we review classic multi-terminal source coding problems: the Slepian–Wolf problem [2], source coding with a helper [7, 8], the Gray–Wyner system [9], and the Kaspi/Heegard–Berger problem [10, 11]. For the first three problems, we introduce extra side information at the receiver, which draws a closer connection with the problem of sequential coding for computing considered in Chapter 3. In the second part, we provide an overview of function computation in information theory. We start with the problem of lossless coding for computing with side information [6] and the Körner–Marton problem [3]. Finally, we review the linear computation coding developed by Nazer and Gastpar [4].

Definition 2.1 (Discrete Memoryless Source)**.** *A* discrete memoryless source *(DMS) $\langle X \rangle$ is specified by a finite alphabet $\mathcal{X}$ and a probability mass function (pmf) p_X over $\mathcal{X}$. The DMS $\langle X \rangle$ generates an independent and identically distributed (i.i.d.) random process $\{X_i\}$ with $X_i \sim p_X$.*

Note that the above definition also includes the multivariate case. For example, if $X = (S_1, S_2, \cdots, S_L)$, then $p_X = p_{S_1,S_2,\cdots,S_L}$, $\mathcal{X} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_L$, and the DMS $\langle S_1, \cdots, S_L \rangle$ generates an i.i.d. random process $\{(S_{1i}, \cdots, S_{Li})\}$.

Definition 2.2 (Doubly Symmetric Binary Source)**.** *Fix $\alpha \in [0, 1/2]$. A doubly symmetric binary source (DSBS(α)) is a DMS $\langle X_1, X_2 \rangle$ with $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$ and*

$$p_{X_1,X_2}(0,0) = p_{X_1,X_2}(1,1) = \frac{1-\alpha}{2},$$
$$p_{X_1,X_2}(0,1) = p_{X_1,X_2}(1,0) = \frac{\alpha}{2}.$$

Equivalently, one can think of a DSBS(α) follows from the following construction: Assume that $S \sim$ Bernoulli(1/2) and $Z \sim$ Bernoulli(α) are independent. Then, $\langle S, S \oplus Z \rangle$ is a DSBS(α).

Definition 2.3 (Discrete Memoryless Channel)**.** *A* discrete memoryless channel *(DMC) $\langle p_{Y|X} \rangle$ is specified by a finite input alphabet $\mathcal{X}$, a finite output alphabet $\mathcal{Y}$ and a conditional pmf $p_{Y|X}$ over $\mathcal{X} \times \mathcal{Y}$. If an input symbol $x \in \mathcal{X}$ is transmitted, the channel outputs the symbol $y \in \mathcal{Y}$ with probability $p_{Y|X}(y|x)$.*

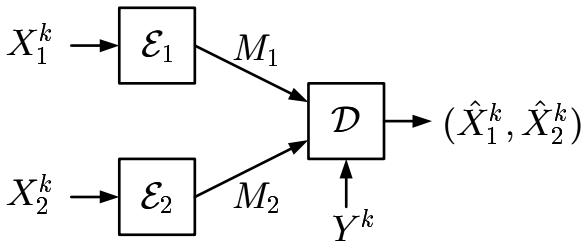

Figure 2.1: The problem of distributed lossless source coding.

2.1 Distributed Lossless Source Coding

A DMS $\langle X_1, X_2, Y\rangle$ generates a source sequence (X_1^k, X_2^k, Y^k), where $k \in \mathbb{Z}^+$. There are two encoding terminals and one decoding terminal. Encoder $j \in \{1, 2\}$ observes the sequence X_j^k and sends a description $M_j \in [2^{kR_j}]$, which is a function of X_j^k, to the decoder. The decoder observes Y^k and wishes to recover (X_1^k, X_2^k) from (M_1, M_2, Y^k) with vanishing error as the length k increases. The system is depicted in Figure 2.1. The goal is to characterize all rate pairs (R_1, R_2) that allow lossless compression.

The optimal rate region $\mathcal{R}^*$ is characterized by Slepian and Wolf in the following theorem.[1]

Theorem 2.1 (Slepian–Wolf)**.** *Consider the problem of distributed lossless source coding. The optimal rate region $\mathcal{R}^*$ is the set of rate pairs (R_1, R_2) such that*

$$\begin{aligned} R_1 &\geq H(X_1|X_2, Y), \\ R_2 &\geq H(X_2|X_1, Y), \\ R_1 + R_2 &\geq H(X_1, X_2|Y). \end{aligned}$$

Theorem 2.1 implies that if the decoder wants to recover the full source, then

1. any cooperation between the encoders cannot lower the sum rate;

2. revealing the side information Y^k to the encoders cannot enlarge the achievable rate region.

Besides the original proof from Slepian and Wolf, a more elegant proof is due to Cover [12], who introduced the random binning argument. Furthermore, Csiszár showed that the binning operation can be realized by random linear codes [13].

Theorem 2.2 (Csiszár)**.** *Fix $\epsilon \in (0, 1]$ and a finite field $\mathbb{F}_q$ with $q \geq \max\{|\mathcal{X}_1|, |\mathcal{X}_2|\}$. For $j \in \{1, 2\}$, let ϕ_j be any bijection between $\mathcal{X}_j$ and $\mathbb{F}_q$. Consider any rate pair (R_1, R_2) in the optimal rate region $\mathcal{R}^*$. For k large enough, there exist matrices $\mathbf{H}_1$*

[1] We say that a rate region is optimal if each of its boundary points is Pareto optimal, i.e., it is impossible to make any one individual entry better off without making at least one individual entry worse off. In this thesis, the optimal rate region of different problems will all be simply denoted by $\mathcal{R}^*$. It will be clear from the context which problem we are referring to.

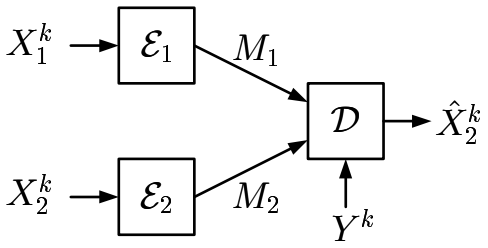

Figure 2.2: The problem of lossless source coding with a helper.

and $\mathbf{H}_2$ *of size* $\lceil \frac{kR_1}{\log q} \rceil \times k$ *and* $\lceil \frac{kR_2}{\log q} \rceil \times k$*, respectively, with entries from* $\mathbb{F}_q$ *and an associated decoding function* $\mathcal{D}(\cdot)$ *such that*

$$\mathbb{P}\left(\mathcal{D}(\mathbf{H}_1\tilde{X}_1^k, \mathbf{H}_2\tilde{X}_2^k, Y^k) \neq (X_1^k, X_2^k)\right) < \epsilon,$$

where $\tilde{X}_j = \phi_j(X_j)$*,* $j \in \{1, 2\}$*.*

Finally, we remark that a multi-user extension of the problem can also be tackled in a similar way. Denote by L the number of encoders. Then, the optimal rate region is the set of rate tuples $(R_1, R_2, \cdots, R_L)$ such that for all $\mathcal{S} \subseteq [L]$,

$$\sum_{\ell \in \mathcal{S}} R_\ell \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}, Y),$$

where $\mathcal{S}^c := [L] \backslash \mathcal{S}$.

2.2 Lossless Source Coding with a Helper

The problem setup of lossless source coding with a helper, as depicted in Figure 2.2, is almost the same as the problem of distributed lossless source coding except that now the decoder is only interested in recovering X_2^k, the sequence observed by Encoder 2. Thus, Encoder 1 serves as a helper to assist Encoder 2 in communicating X_2^k to the decoder. The following theorem gives a single-letter characterization of the optimal rate region. A proof of Theorem 2.3 can be found in [5, Chapter 10.4].

Theorem 2.3 (Wyner/Ahlswede–Körner)**.** *Consider the problem of lossless source coding with a helper. The optimal rate region* $\mathcal{R}^*$ *is the set of rate pairs* (R_1, R_2) *such that*

$$R_1 \geq I(X_1; V|Y),$$
$$R_2 \geq H(X_2|V, Y),$$

for some conditional pmf $p_{V|X_1}$ *with* $|\mathcal{V}| \leq |\mathcal{X}_1| + 1$*.*

However, different from the problem of distributed lossless source coding, in this problem setup revealing the side information Y^k to Encoder 1 may enlarge the achievable rate region. One example is the problem of sequential coding for computing considered in Chapter 3. Besides, if Encoder 1 wants to participate in the communication, then in general information redundancy is inevitable. To see this,

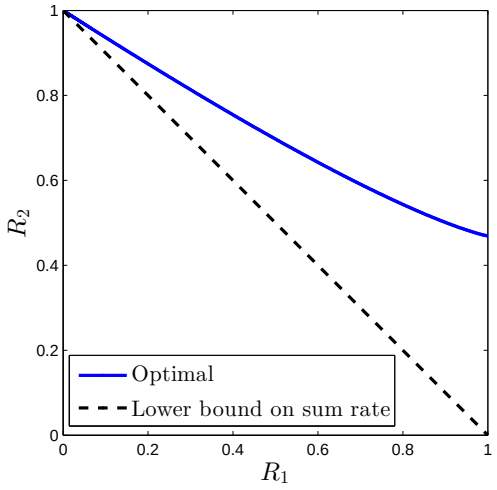

Figure 2.3: An example of source coding with a helper with $(X_1, X_2) \sim \mathsf{DSBS}(0.1)$ and $Y = \emptyset$. The optimal rate region $\mathcal{R}^*$ is plotted in blue solid and the lower bound on sum rate is plotted in black dash.

consider the case where X_1 and (X_2, Y) have an indecomposable joint distribution [14, Problem 15.12]. The optimal sum rate can be lower bounded as

$$\begin{aligned} R_1 + R_2 &\geq I(X_1; V|Y) + H(X_2|V, Y) \\ &= I(X_1; V|Y) + H(X_2|Y) - I(X_2; V|Y) \\ &\overset{(a)}{=} I(X_1, X_2; V|Y) + H(X_2|Y) - I(X_2; V|Y) \\ &= H(X_2|Y) + I(X_1; V|X_2, Y) \\ &\geq H(X_2|Y), \end{aligned}$$

where (a) follows since $V \multimap\!\!- X_1 \multimap\!\!- (X_2, Y)$ form a Markov chain. Therefore, in order to attain the minimum sum rate, it requires that $I(X_1; V|X_2, Y) = 0$, i.e., $V \multimap\!\!- (X_2, Y) \multimap\!\!- X_1$ form a Markov chain. Since X_1 and (X_2, Y) have an indecomposable joint distribution, the two Markov chains imply that V is independent of (X_1, X_2, Y) [14, Problem 16.25]. That is, to achieve the minimum sum rate Encoder 2 cannot accept any help from Encoder 1.

Example 2.1. *Let $\langle X_1, X_2 \rangle$ be a DSBS(α) and $Y = \emptyset$. For this case, the optimal rate region $\mathcal{R}^*$ is characterized by Wyner [15] as the set of rate pairs (R_1, R_2) such that*

$$\begin{aligned} R_1 &\geq 1 - h_\mathsf{b}(\beta), \\ R_2 &\geq h_\mathsf{b}(\beta(1-\alpha) + (1-\beta)\alpha), \end{aligned}$$

for some $\beta \in [0, 1/2]$. As shown in Figure 2.3, the minimum sum rate can only be attained at $R_1 = 0$.

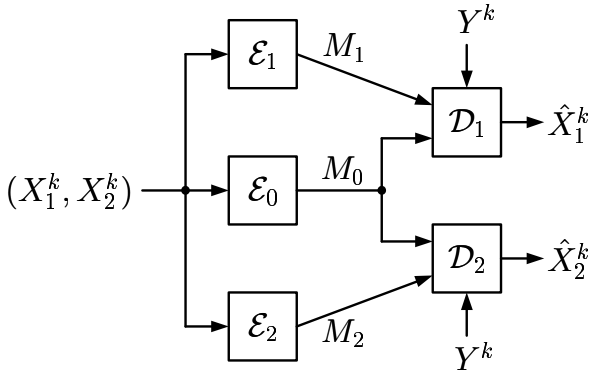

Figure 2.4: The Gray–Wyner system.

2.3 Gray–Wyner System

A DMS $\langle X_1, X_2, Y\rangle$ generates a source sequence (X_1^k, X_2^k, Y^k). There are three encoding terminals and two decoding terminals. The encoders observe the sequence (X_1^k, X_2^k) and the decoders observe the sequence Y^k. Encoder $j \in \{1,2\}$ sends a description $M_j \in [2^{kR_j}]$ to Decoder j and Encoder 0 sends a description $M_0 \in [2^{kR_0}]$ to both decoders. The messages M_0, M_1, M_2 are functions of (X_1^k, X_2^k). Decoder $j \in \{1,2\}$ observes Y^k and wishes to recover X_j^k from (M_0, M_j, Y^k) with vanishing error as the length k increases. The system is plotted in Figure 2.4. The goal is to characterize all rate triples (R_0, R_1, R_2) that admit lossless compression. The following theorem gives a single-letter characterization of the optimal rate region.

Theorem 2.4 (Gray–Wyner)**.** *Consider the Gray–Wyner system. The optimal rate region $\mathcal{R}^*$ is the set of rate triples (R_0, R_1, R_2) such that*

$$\begin{aligned} R_0 &\geq I(X_1, X_2; V|Y), \\ R_1 &\geq H(X_1|V, Y), \\ R_2 &\geq H(X_2|V, Y), \end{aligned}$$

for some conditional pmf $p_{V|X_1,X_2}$ with $|\mathcal{V}| \leq |\mathcal{X}_1||\mathcal{X}_2| + 2$.

From Theorem 2.4, we observe that it is optimal that Encoder 0 acts as a helper for Encoders 1 and 2. The optimal sum rate can be lower bounded as

$$\begin{aligned} R_0 + R_1 + R_2 &\geq I(X_1, X_2; V|Y) + H(X_1|V, Y) + H(X_2|V, Y) \\ &= H(X_1, X_2|Y) + I(X_1; X_2|V, Y) \\ &\geq H(X_1, X_2|Y). \end{aligned}$$

As can be seen, in order to attain the minimum sum rate $H(X_1, X_2|Y)$, the common rate R_0 should be at least as large as

$$\min_{\substack{p_{V|X_1,X_2} \\ \text{s.t. } I(X_1;X_2|V,Y)=0}} I(X_1, X_2; V|Y),$$

which is known as Wyner's common information [16] when $Y = \emptyset$. For other notions of common information related to the Gray–Wyner system, we refer the reader to [5, Chapter 14.2.1].

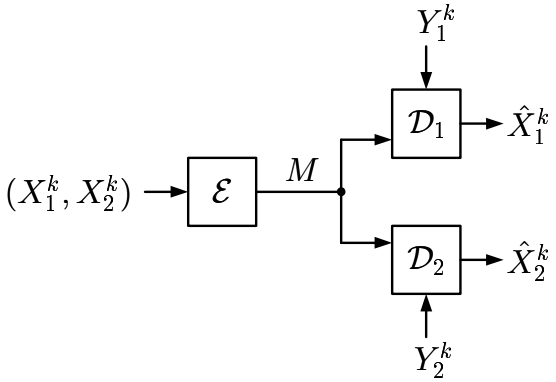

Figure 2.5: The Kaspi/Heegard–Berger Problem.

2.4 Kaspi/Heegard–Berger Problem

Here we present a lossless version of the Kaspi/Heegard–Berger Problem. A DMS $\langle X_1, X_2, Y_1, Y_2 \rangle$ generates a source sequence $(X_1^k, X_2^k, Y_1^k, Y_2^k)$. There are one encoding terminal and two decoding terminals. The encoder observes the sequences (X_1^k, X_2^k) and Decoder $j \in \{1, 2\}$ has side information Y_j^k. The encoder sends a description $M \in [2^{kR}]$, a function of (X_1^k, X_2^k), to both decoders. Decoder $j \in \{1, 2\}$ wishes to recover X_j^k from (M, Y_j^k) with vanishing error as the length k increases. The system is plotted in Figure 2.5. The goal is to characterize the minimum compression rate. Unfortunately, this problem remains open in general. Nevertheless, for the following special cases, the optimal rates are known:

1. $X_1 = X_2$ [17];

2. X_1 is a function of X_2, or vice versa [18];

3. the side information is physically degraded [11];

4. the side information is conditionally less noisy [19].

The case where the encoder also knows the side information was studied in [20, 21]. The problem has also been extended to multiple decoders [22]. To conclude this section, we present an achievable compression rate, which follows by first sending a common description U and then performing the Slepian–Wolf coding for individual destinations.

Proposition 2.1 ([21, Lemma 1])**.** *Consider the (lossless) Kaspi/Heegard–Berger problem. The optimal compression rate R^* is upper bounded by*

$$R^* \leq \min_{p_{U|X_1,X_2}} \left\{ \max_{j\in\{1,2\}} \{I(X_1, X_2; U|Y_j)\} + H(X_1|U, Y_1) + H(X_2|U, Y_2) \right\},$$

where $|\mathcal{U}| \leq |\mathcal{X}_1||\mathcal{X}_2| + 3$.

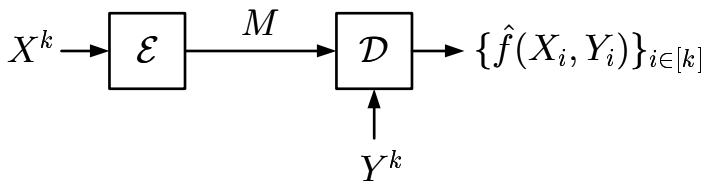

Figure 2.6: The problem of lossless coding for computing.

2.5 Lossless Coding for Computing

We now review the problem of lossless coding for computing with side information [6] (see Figure 2.6). There is a DMS $\langle X, Y\rangle$ which generates a source sequence (X^k, Y^k). The encoder observes the sequence X^k and sends a message $M \in [2^{kR}]$ to the decoder. The decoder observes the sequence Y^k and wishes to recover an element-wise function $f(x, y)$ with vanishing error as k increases. The goal is to characterize the minimum compression rate such that the decoder can recover the desired function losslessly. The following theorem gives a single-letter characterization of the optimal compression rate.

Theorem 2.5 (Orlitsky–Roche)**.** *Consider the problem of lossless coding for computing. The optimal rate R^* can be expressed as*

$$R^* = \min_{\substack{p_{V|X} \\ s.t.\ H(f(X,Y)|V,Y)=0}} I(X; V|Y),$$

with $|\mathcal{V}| \le |\mathcal{X}| + 1$.

We remark that the optimal compression rate can be further refined as the conditional graph entropy [6] (see also [5, Chapter 21.1]). Here we provide a proof of Theorem 2.5 without the details of the analysis of error probability, which follows essentially the same lines as [5, Chapter 11.3]. Our main goal is to review the standard random coding arguments for achievability: joint typicality encoding, binning, and joint typicality decoding and the standard procedure for the converse proof, which will be heavily used in Chapters 3 and 4.

Proof: For convenience, denote $s_i = f(x_i, y_i)$, $i \in [k]$.

(Achievability.) Fix the conditional pmf $p_{V|X}$ such that $H(f(X, Y)|V, Y) = 0$. Let $p_V(v) = \sum_{x\in\mathcal{X}} p_X(x)p_{V|X}(v|x)$, for all $v \in \mathcal{V}$. Since $H(f(X, Y)|V, Y) = 0$, there exists a function g such that $g(V, Y) = f(X, Y)$ almost surely.

Codebook generation: Randomly and independently generate $\lceil 2^{kR}\rceil\lceil 2^{k\tilde{R}}\rceil$ sequences $v^k(m, \ell)$, $m \in [2^{kR}]$, $\ell \in [2^{k\tilde{R}}]$, each according to $\prod_{i=1}^k p_V(v_i)$. The codebook is revealed to both the encoder and the decoder.

Encoding: Upon seeing the sequence x^k, the encoder finds an index pair (m, ℓ) such that $(x^k, v^k(m, \ell)) \in T_{\epsilon'}^{(k)}(X, V)$. If there is more than one such index, it selects the one that minimizes $m\lceil 2^{k\hat{R}}\rceil + \ell$. If there is no such index, it sets $(m, \ell) = (1, 1)$. Then, the encoder sends the index m to the decoder.

Decoding: Let $\epsilon > \epsilon'$. Upon seeing the index m, the decoder finds the unique index $\hat{\ell}$ such that $(v^k(m, \hat{\ell}), y^k) \in \mathcal{T}_{\epsilon}^{(k)}(V, Y)$; otherwise it sets $\hat{\ell} = 1$. The decoder then computes the reconstruction sequence $\hat{s}_i = g(v_i(m, \hat{\ell}), y_i)$ for all $i \in [k]$.

We skip the analysis of error probability, the details of which can be found in [5, Chapter 11.3.1]. Finally, if it holds that $(V^k(M,L), X^k, Y^k) \in T_\epsilon^{(k)}(V,X,Y)$, then from the union bound and the typical average lemma we have

$$\begin{aligned}
&\mathbb{P}\left(\bigcup_{i=1}^{k}\{g(v_i,Y_i)\neq f(X_i,Y_i)\}\Big|(v^k,X^k,Y^k)\in\mathcal{T}_\epsilon^{(k)}\right)\\
&\leq \sum_{i=1}^{k}\mathbb{P}\left(\{g(v_i,Y_i)\neq f(X_i,Y_i)\}\Big|(v^k,X^k,Y^k)\in\mathcal{T}_\epsilon^{(k)}\right)\\
&\leq k(1+\epsilon)\mathbb{P}(g(V,Y)\neq f(X,Y))\\
&= 0.
\end{aligned}$$

(Converse.) Denote by M the message sent by the encoder and by $\hat{S}^k$ the estimate of S^k. Let $Q\sim$Uniform$([k])$ be independent of (X^k,Y^k). First, we have

$$\begin{aligned}
kR &\geq H(M|Y^k)\\
&= I(X^k;M|Y^k)\\
&= \sum_{i=1}^{k} I(X_i;M|X^{i-1},Y^k)\\
&\overset{(a)}{=} \sum_{i=1}^{k} I(X_i;M,X^{i-1},Y^{[k]\setminus\{i\}}|Y_i)\\
&\overset{(b)}{=} \sum_{i=1}^{k} I(X_i;V_i|Y_i),\\
&\overset{(c)}{=} \sum_{i=1}^{k} I(X_i;V_i|Y_i,Q=i),\\
&= kI(X_Q;V_Q|Y_Q,Q),\\
&\overset{(d)}{=} kI(X_Q;V_Q,Q|Y_Q),
\end{aligned}$$

where (a) follows since $(X^{i-1},Y^{[k]\setminus\{i\}})$ $-\!\circ\!-$ Y_i $-\!\circ\!-$ X_i form a Markov chain and (c) and (d) follow since Q is independent of (X^k,Y^k). For the step (b), we define $V_i=(M,X^{i-1},Y^{[k]\setminus\{i\}})$. Next, denoting $\mathsf{P}_\mathsf{e}^{(k)}=\mathbb{P}(\hat{S}^k\neq S^k)$, we have

$$\begin{aligned}
h_\mathsf{b}(\mathsf{P}_\mathsf{e}^{(k)})+k\mathsf{P}_\mathsf{e}^{(k)}\log(|\mathcal{S}|) &\overset{(a)}{\geq} H(S^k|\hat{S}^k)\\
&\overset{(b)}{\geq} H(S^k|M,Y^k)\\
&= \sum_{i=1}^{k} H(S_i|M,S^{i-1},Y^k)\\
&\geq \sum_{i=1}^{k} H(S_i|V_i,Y_i),\\
&= \sum_{i=1}^{k} H(S_i|V_i,Y_i,Q=i),\\
&= kH(S_Q|V_Q,Q,Y_Q),
\end{aligned}$$

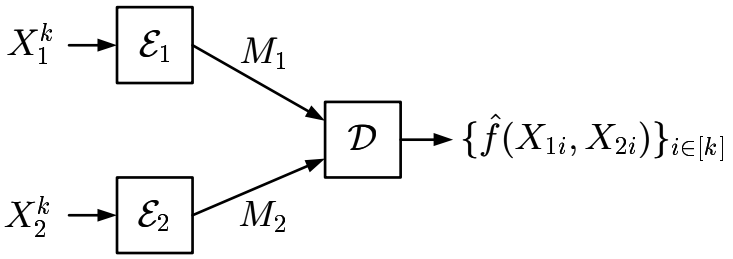

Figure 2.7: The problem of distributed lossless computing.

where (a) follows from Fano's inequality and (b) follows from the data processing inequality. Note that $(X_Q, Y_Q) \sim p_{X,Y}$ and that $(V_Q, Q) \multimap\!\!- X_Q \multimap\!\!- Y_Q$ form a Markov chain. Thus, we identify $(X, Y, S) = (X_Q, Y_Q, S_Q)$ and set $V = (V_Q, Q)$ to obtain

$$\begin{aligned} R &\geq I(X; V|Y), \\ H(f(X,Y)|V,Y) &\leq h_{\mathsf{b}}(\mathsf{P}_{\mathsf{e}}^{(k)})/k + \mathsf{P}_{\mathsf{e}}^{(k)} \log(|\mathcal{S}|). \end{aligned}$$

Finally, by the assumption that $\lim_{k\to\infty} \mathsf{P}_{\mathsf{e}}^{(k)} = 0$, the converse is established by letting $k \to \infty$. ∎

Hereafter, we use the shorthand notation $\epsilon_k := h_{\mathsf{b}}(\mathsf{P}_{\mathsf{e}}^{(k)})/k + \mathsf{P}_{\mathsf{e}} \log|\mathcal{A}|$, where the set $\mathcal{A}$ will be clear from the context.

2.6 Körner–Marton Problem

Consider the problem of distributed lossless computing depicted in Figure 2.7. In this problem, the decoder only wishes to recover an element-wise function $f(x_1, x_2)$ losslessly rather than the entire source sequences. This problem remains open in general. Now let us consider a special case studied by Körner and Marton: $\langle X_1, X_2 \rangle$ is a DSBS(α), where $\alpha \in [0, 1]$, and the desired function is the modulo-two sum $f(x_1, x_2) = x_1 \oplus x_2$. For all previous problems with known optimality results, we can use a standard random coding argument to establish the achievability. However, for this particular instance, it is only known that the optimal compression rate can be achieved by linear codes, which constitute a class of codes with *structures*.

Theorem 2.6 (Körner–Marton)**.** *Consider the problem of distributed lossless computing. Let $\langle X_1, X_2 \rangle \sim$ DSBS(α), where $\alpha \in [0, 1]$, and $f(x_1, x_2) = x_1 \oplus x_2$. Then, the optimal rate region $\mathcal{R}^*$ is the set of rate pairs (R_1, R_2) such that*

$$\begin{aligned} R_1 &\geq h_{\mathsf{b}}(\alpha), \\ R_2 &\geq h_{\mathsf{b}}(\alpha). \end{aligned}$$

Proof: (Converse.) The converse follows from evaluating the outer bound: $R_1 \geq H(f(X_1, X_2)|X_2)$ and $R_2 \geq H(f(X_1, X_2)|X_1)$. Note that $X_1 \oplus X_2$ is independent of each X_j, $j \in \{1, 2\}$.

(Achievability.) For notational convenience, denote $Z = X_1 \oplus X_2$. First, note that Theorem 2.2 implies that there exist good linear codes for point-to-point source

coding. Denote by $\mathbf{H}$ the compression matrix of size $kR \times k$ for the sequence Z^k. For the decoder to recover Z^k losslessly, it requires that $R \geq H(Z) = h_{\mathsf{b}}(\alpha)$. The compression matrix $\mathbf{H}$ is revealed to all nodes. Upon seeing x_j^k, $j \in \{1, 2\}$, Encoder j sends the message $M_j = \mathbf{H}x_j^k$ to the decoder. Upon seeing M_1 and M_2, the decoder first computes the bit-wise modulo-two sum $(\mathbf{H}x_1^k) \oplus (\mathbf{H}x_2^k) = \mathbf{H}z^k$ and then use the typicality decoding to recover z^k. Thus, any rate pair (R_1, R_2) satisfying $R_j \geq h_{\mathsf{b}}(\alpha)$, $j \in \{1, 2\}$ is achievable. ■

From Theorem 2.6, we can observe the following. First, decoding the full sequences results in the sum rate $1 + h_{\mathsf{b}}(\alpha)$, which is much higher than the optimal sum rate when $\alpha \ll 1$. Second, the entropy of the desired information is $h_{\mathsf{b}}(\alpha)$, but the required sum rate is $2h_{\mathsf{b}}(\alpha)$. Note that for each $j \in \{1, 2\}$, X_j and $X_1 \oplus X_2$ are independent and thus M_j itself only contains unwanted information. Due to lack of coordination, this kind of redundancy is inevitable.

Recall that we can think of that $(X_1, X_2) = (S, S \oplus Z)$, where $S \sim \text{Bernoulli}(1/2)$ and $Z \sim \text{Bernoulli}(\alpha)$ are independent. Thus, the modulo-2 sum is more like a difference between X_1 and X_2, instead of a sum. Furthermore, if S is not uniformly distributed, the superiority of the Körner–Marton coding may disappear. The following example is a demonstration.

Example 2.2. *Let $S, Z \in \mathbb{F}_q$ be two independent random variables, where $q > 2$. Assume that $X_1 = S$ and $X_2 = S \oplus_q Z$. Let us first consider computation of the function $f(x_1, x_2) = x_2 \ominus_q x_1$. In this case, the Slepian–Wolf coding achieves the sum rate $H(X_1, X_2) = H(S) + H(Z)$ and the Körner–Marton coding achieves the sum rate $2H(X_2 \ominus_q X_1) = 2H(Z)$. Thus, if $H(S) < H(Z)$, then the Slepian–Wolf coding outperforms the Körner–Marton coding. The reason is that in the Körner–Marton coding, Encoder 1 has to add redundancy in representation to maintain the linear structure.*

Next, consider computation of the function $f(x_1, x_2) = x_1 \oplus_q x_2$. In this case, the Slepian–Wolf coding again achieves the sum rate $H(X_1, X_2) = H(S) + H(Z)$ and the Körner–Marton coding achieves the sum rate $2H(X_1 \oplus_q X_2) = 2H(S \oplus_q S \oplus_q Z)$. Thus, the Slepian–Wolf coding always outperforms the Körner–Marton coding. Similarly, in the Körner–Marton coding, both encoders have to add redundancy in representation to maintain the linear structure.

Finally, combining the distributed compress–bin scheme [23, 24] and the Körner–Marton coding, Ahlswede and Han [25, Section VI] established the following achievability. They showed through an example that the resulting rate region can be strictly larger than the union of the individual achievable rate regions of the distributed compress–bin scheme and the Körner–Marton coding.

Proposition 2.2 (Ahlswede–Han)**.** *Consider the problem of distributed lossless computing. Assume that $X_1, X_2 \in \mathbb{F}_q$ and $f(x_1, x_2) = x_1 \oplus_q x_2$. A rate pair (R_1, R_2) is achievable if*

$$\begin{aligned} R_1 &> I(V_1; X_1|V_2) + H(X_1 \oplus_q X_2|V_1, V_2), \\ R_2 &> I(V_2; X_2|V_1) + H(X_1 \oplus_q X_2|V_1, V_2), \\ R_1 + R_2 &> I(V_1, V_2; X_1, X_2) + 2H(X_1 \oplus X_2|V_1, V_2), \end{aligned}$$

for some conditional pmf $p_{V_1,V_2|X_1,X_2}$ satisfying the Markov chain $V_1 \multimap X_1 \multimap X_2 \multimap V_2$.

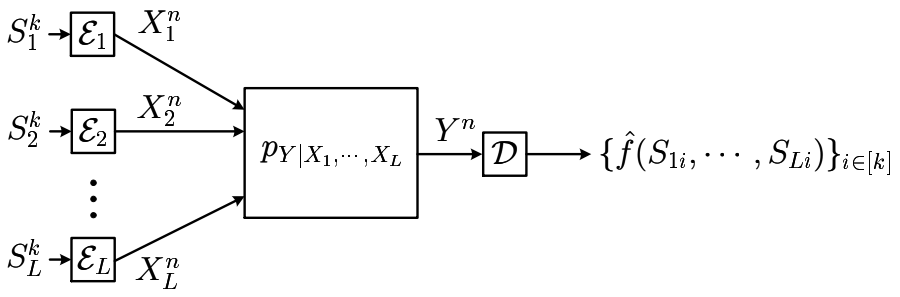

Figure 2.8: The problem of function computation over a MAC.

2.7 Linear Computation Coding

A discrete memoryless L-input MAC $\langle p_{Y|X_1,\cdots,X_L}\rangle$ is a DMC governed by the conditional pmf $p_{Y|X_1,\cdots,X_L}$ with multiple distributed channel inputs. The problem of function computation over a MAC is depicted in Figure 2.8. The destination wishes to recover an element-wise function $f(s_1,\cdots,s_L)$ of the distributed sequences $(S_1^k,\cdots,S_L^k)$ generated by a DMS $\langle S_1,\cdots,S_L\rangle$. The problem statement, including the definitions of computation code, computation rate, and computation capacity, can be found in Section 6.1.

Linear computation coding is a coding technique developed by Nazer and Gastpar [4], which computes linear functions efficiently over MACs with a matched linear structure.

Definition 2.4. *Let $\mathbb{F}_q$ be a finite field. An L-input MAC $\langle p_{Y|X_1,\cdots,X_L}\rangle$ is called $\mathbb{F}_q$-linear if $\mathcal{X}_j=\mathbb{F}_q$ and there exists a random variable W such that*

$$W=\bigoplus_{\ell=1}^{L} a_\ell\otimes X_\ell,$$

where $a_\ell\in\mathbb{F}_q$ for all $\ell\in[L]$ and $p_{Y|W,X_1,\cdots,X_L}=p_{Y|W}$. Similarly, a function $f(s_1,\cdots,s_L)$ is called $\mathbb{F}_q$-linear if there exists a vector $b_{[L]}\in\mathbb{F}_q^L$ such that

$$f(s_1,\cdots,s_L)=\bigoplus_{\ell=1}^{L} b_\ell\otimes s_\ell,$$

Definition 2.5. *We say that a DMC is* symmetric *if the output symbols can be placed into subsets such that for each subset the probability transition matrix satisfies that*

1. *each row is a permutation of every other row;*
2. *each column is a permutation of every other column.*

Theorem 2.7 (Nazer–Gastpar)**.** *Let $\langle S_1,\cdots,S_L,V\rangle$ be a DMS. Consider computing an element-wise $\mathbb{F}_q$-linear function $f(s_1,\cdots,s_L)$ over the MAC $\langle p_{Y|X_1,\cdots,X_L}\rangle$ with decoder side information V. If the MAC is $\mathbb{F}_q$-linear, then any computation rate R satisfying*

$$R<\frac{I(W;Y)}{H(f(S_1,\cdots,S_L)|V)}=:R_{\mathsf{sym}}$$

is achievable, where $W \sim$ *Uniform*$(\mathbb{F}_q)$*. Furthermore, If the* $\mathbb{F}_q$*-linear MAC is also symmetric, then* R_{sym} *is the computation capacity.*

The achievability proof of Theorem 2.7 relies on linear binning (see Theorem 2.2) and is in the same spirit of the Körner–Marton coding.

Sequential Coding for Computing – The Single-User Case

3

In a content delivery network, every communication involves at least two steps: *request* and *delivery*.[1] Each end user first sends a request to the servers and then the servers deliver the user's desired content to the end user. From the everyday experience, at some time of the day, usually in the evening, the servers receive more requests than in the other time periods, e.g., in the early morning. However, the available communication resources are more or less evenly distributed over time. Therefore, end users suffer from network congestion during peak-traffic times.

Caching has recently drawn a lot of attention due to its potential to reduce congestion and the experienced delay of end users. Caching is a technique for reallocating communication resources which works as follows. First, local servers and/or end users deploy a special-purpose memory called *cache*. Based on the collected statistics of users' preferences, the servers know which data (files, images, video, audio, etc.) are most popular. Then, one can develop a caching policy to store some compressed version of the database in the available caches. If the caching policy is well-designed, then it should be the case that most of the time the users' desired data is already in their caches or can be directly served by the local servers.

Thus, content delivery with caching involves three stages: *cache*, *request*, and *update*, where the original delivery stage is further divided into the cache and update stages. In the cache stage, the servers have no knowledge of users' requests other than their statistics. As the requests are unknown in the cache stage, redundancy in the cache content is in general inevitable. However, we can still exploit the request statistics to increase the chance of being useful. For most applications, requests require much less communication resources compared to the requested data. Hence, we consider a simplified model which involves only the cache stage and the update stage.

To model a content delivery network, we identify its three key ingredients: database, requests, and the requested data. Then, the database is modeled by a

[1]The material of this chapter has appeared in

1. C.-Y. Wang, S. H. Lim, and M. Gastpar, "Information-theoretic caching," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Hong Kong, China, Jun. 2015.

2. C.-Y. Wang, S. H. Lim, and M. Gastpar, "Information-theoretic caching: Sequential coding for computing," in *arXiv:1504.00553[cs.IT]*, Apr. 2015.

generic random variable $\underline{X}$, the request is modeled by a generic random variable $\underline{Y}$, and the requested data is modeled by a function g of the database $\underline{X}$ and the request $\underline{Y}$, i.e., $g(\underline{X}, \underline{Y})$.

Depending on the targeted applications, there are various ways to model $\underline{X}$, $\underline{Y}$, and g. To name a few, we have

1. one-shot model: $\underline{X} = X$, $\underline{Y} = Y$, and $g(\underline{X}, \underline{Y}) = f(X, Y)$;
2. single-request model: $\underline{X} = X^k$, $\underline{Y} = Y$, and $g(\underline{X}, \underline{Y}) = \{f(X_i, Y)\}_{i \in [k]}$;
3. multi-request model: $\underline{X} = X^k$, $\underline{Y} = Y^k$, and $g(\underline{X}, \underline{Y}) = \{f(X_i, Y_i)\}_{i \in [k]}$.

Here X^k and Y^k are sequences of i.i.d. random variables.

The one-shot model is suitable for delay-sensitive applications which cannot afford coding over multiple instances. The single-request model fits well with applications in which the users' requests remain fixed over the entire time period of interest, e.g., on-demand video streaming of a movie from a given database. Ample results are available for the single-request model at this point: the worst-case analysis [26], the average-case analysis [27], decentralized [28], delay-sensitive [29], online [30], multiple layers [31], request of multiple items [32], secure delivery [33], wireless networks [34, 35], etc. In addition, some improved order-optimal results for the average case can be found in [36, 37].

By contrast, the multi-request model may be an interesting fit for applications in which the users' requests change over time, such as multi-view video systems which offer the flexibility that the users can freely switch camera views. Furthermore, the multi-request model fits well with sensor network applications. In many cases, only the sensor data (modeled as X) with certain properties (modeled as Y) are of interest and the desired properties may vary over a timescale of minutes, hours, or even days. In the following, except Section 3.7, we will restrict attention to the average-case analysis of the multi-request model.

Chapter Outline: This chapter is devoted to the single-user case. In Section 3.1, a formal problem statement is given. Next, we present the optimal rate region and its various properties in Section 3.2. In Sections 3.3, 3.4, 3.5, we consider three special cases: independent source components, nested source components, and arbitrarily correlated components with uniform requests, respectively. In Section 3.6, we discuss the distributed compress–bin scheme with an emphasis on successive decoding. Finally, in Section 3.7 we discuss the single-request model and consider three performance criteria: compound, outage, and adaptive coding.

3.1 Problem Statement

In this and the following chapter, we consider the problem of sequential coding for computing. The single-user case is formulated as follows. A DMS $\langle X, Y \rangle$ generates i.i.d. source sequences (X^k, Y^k). There are two encoding terminals and one decoding terminal. The *cache* encoder observes the source sequence X^k, the *update* encoder observes the source sequences (X^k, Y^k), and the decoder observes Y^k. The cache encoder generates a message $M_{\mathsf{c}} \in [2^{kR_{\mathsf{c}}}]$, and the update encoder generates a message $M_{\mathsf{u}} \in [2^{kR_{\mathsf{u}}}]$. The decoder receives $(M_{\mathsf{c}}, M_{\mathsf{u}})$ and wishes to recover an element-wise function $f(x, y)$ losslessly. The messages M_{c} and M_{u} are referred to

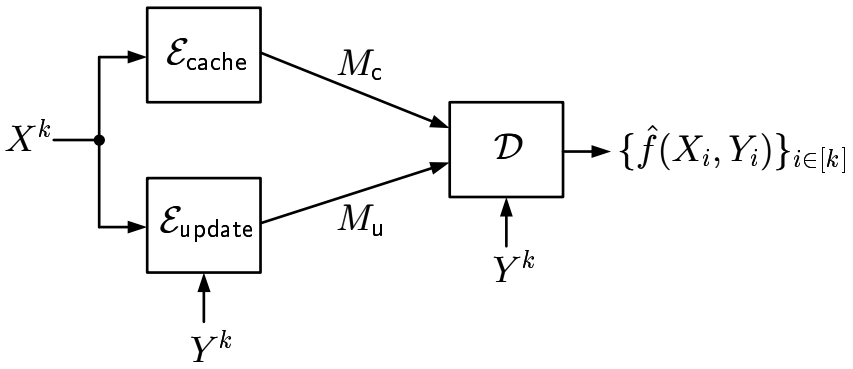

Figure 3.1: The problem of sequential coding for computing: the single-user case.

as *cache* and *update*, respectively. The system is depicted in Figure 3.1. The stated problem will be referred to as *sequential coding for computing*. The name comes from the two related source coding problems: sequential coding of correlated sources [38] and coding for computing [6].

A $(2^{kR_{\mathsf{c}}}, 2^{kR_{\mathsf{u}}}, k)$ distributed multiple description code consists of

- two encoders, where the cache encoder assigns an index $m_{\mathsf{c}}(x^k) \in [2^{kR_{\mathsf{c}}}]$ to each sequence $x^k \in \mathcal{X}^k$ and the update encoder assigns an index $m_{\mathsf{u}}(x^k, y^k) \in [2^{kR_{\mathsf{u}}}]$ to each pair of sequences $(x^k, y^k) \in \mathcal{X}^k \times \mathcal{Y}^k$;
- one decoder, which assigns an estimate $\hat{s}^k$ to each tuple $(m_{\mathsf{c}}, m_{\mathsf{u}}, y^k)$.

We say that a rate pair $(R_{\mathsf{c}}, R_{\mathsf{u}})$ is achievable if there exists a sequence of $(2^{kR_{\mathsf{c}}}, 2^{kR_{\mathsf{u}}}, k)$ codes with

$$\lim_{k\to\infty} \mathbb{P}\left(\bigcup_{i\in[k]} \left\{\hat{S}_i \neq f(X_i, Y_i)\right\}\right) = 0. \tag{3.1}$$

The optimal rate region $\mathcal{R}^*$ is the closure of the set of achievable rate pairs.

3.2 The Optimal Rate Region and its Properties

As can be seen from Figure 3.1, this setup is closely related to the problem of lossless source coding with a helper. That is, the cache itself can be considered as a helper (see Section 2.2), which helps reducing the required rate for the update. The following theorem provides a single-letter characterization of the optimal rate region $\mathcal{R}^*$.

Theorem 3.1. *The optimal rate region $\mathcal{R}^*$ is the set of rate pairs $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that*

$$R_{\mathsf{c}} \geq I(X; V|Y), \tag{3.2}$$
$$R_{\mathsf{u}} \geq H(f(X, Y)|V, Y), \tag{3.3}$$

for some conditional pmf $p_{V|X}$ with $|\mathcal{V}| \leq |\mathcal{X}| + 1$.

Proof: (Achievability.) The proof follows from standard random coding arguments as in the problem of lossless source coding with a helper. Here we provide a high-level description of the coding scheme. First, the cache encoder applies Wyner–Ziv coding on x^k assuming decoder side information y^k so that the decoder learns v^k, a quantized version of x^k. Then the update encoder applies Slepian–Wolf coding on $\{f(x_i, y_i)\}_{i\in[k]}$ assuming decoder side information (v^k, y^k).

(Converse.) Denote $S_i = f(X_i, Y_i)$ for all $i \in [k]$. First, we have

$$\begin{aligned} kR_{\mathsf{c}} &\geq H(M_{\mathsf{c}}) \\ &\geq H(M_{\mathsf{c}}|Y^k) \\ &= I(X^k; M_{\mathsf{c}}|Y^k) \\ &= \sum_{i=1}^{k} I(X_i; M_{\mathsf{c}}|X^{i-1}, Y^k) \\ &\overset{(a)}{=} \sum_{i=1}^{k} I(X_i; M_{\mathsf{c}}, X^{i-1}, Y^{[k]\setminus\{i\}}|Y_i) \\ &\geq \sum_{i=1}^{k} I(X_i; V_i|Y_i), \end{aligned}$$

where (a) follows since (X_i, Y_i) is independent of $(X^{i-1}, Y^{[k]\setminus\{i\}})$. For the last step, we define $V_i = (M_{\mathsf{c}}, S^{i-1}, Y^{[k]\setminus\{i\}})$. Note that $V_i \multimap X_i \multimap Y_i$ form a Markov chain.

Next, we have

$$\begin{aligned} kR_{\mathsf{u}} &\geq H(M_{\mathsf{u}}) \\ &\geq H(M_{\mathsf{u}}|M_{\mathsf{c}}, Y^k) \\ &= H(S^k, M_{\mathsf{u}}|M_{\mathsf{c}}, Y^k) - H(S^k|M_{\mathsf{c}}, M_{\mathsf{u}}, Y^k) \\ &\overset{(a)}{\geq} H(S^k|M_{\mathsf{c}}, Y^k) - k\epsilon_k \\ &= \sum_{i=1}^{k} H(S_i|M_{\mathsf{c}}, S^{i-1}, Y^k) - k\epsilon_k \\ &= \sum_{i=1}^{k} H(S_i|V_i, Y_i) - k\epsilon_k, \end{aligned}$$

where (a) follows from the data processing inequality and Fano's inequality. The rest of the proof follows from the standard time-sharing argument and then letting $k \to \infty$. The cardinality bound on $\mathcal{V}$ can be proved using the convex cover method [5, Appendix C]. ∎

We remark that the converse can also be established by identifying the auxiliary random variable $V_i = (M_{\mathsf{c}}, X^{i-1}, Y^{[k]\setminus\{i\}})$. The optimal rate region $\mathcal{R}^*$ is convex since the auxiliary random variable V performs time sharing implicitly. As can be seen from the achievability, even if the update encoder is restricted to access only the sequence of functions $\{f(x_i, y_i)\}_{i\in[k]}$, instead of (x^k, y^k), the rate region remains the same.

A simple consequence of Theorem 3.1 is the following lower bound on the sum rate.

Corollary 3.1. *$R_{\mathsf{c}} + R_{\mathsf{u}} \geq H(f(X,Y)|Y)$ for all $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*$.*

Proof: The corollary can be proved by a simple cut-set argument. Alternatively, from Theorem 3.1 we observe that

$$\begin{aligned} R_{\mathsf{c}} + R_{\mathsf{u}} &\geq I(X;V|Y) + H(f(X,Y)|V,Y) \\ &\geq I(f(X,Y);V|Y) + H(f(X,Y)|V,Y) \\ &= H(f(X,Y)|Y). \end{aligned}$$

■

Let us briefly mention the two extreme cases. First, when $R_{\mathsf{c}} = 0$, the optimal update rate is $R_{\mathsf{u}}^* = H(f(X,Y)|Y)$. Second, when the update rate $R_{\mathsf{u}} = 0$, the minimum cache rate is

$$R_{\mathsf{c}}^* = \min_{\substack{p_{V|X} \\ \text{s.t. } H(f(X,Y)|V,Y)=0}} I(X;V|Y).$$

Thus, we recover the result of lossless coding for computing with side information (see Section 2.5). Note that $R_{\mathsf{u}}^* \leq R_{\mathsf{c}}^*$ and the inequality can be strict. Therefore, there is a penalty when the request Y^k is not known at the encoder.

Due to the fact that the update encoder is more informative than the cache encoder, we have the following two corollaries regarding in which direction we can move a partial rate such that the resulting rate pair still resides in $\mathcal{R}^*$.

Corollary 3.2. *If $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*$, then for all $a \in [0, R_{\mathsf{c}}]$, $(R_{\mathsf{c}} - a, R_{\mathsf{u}} + a) \in \mathcal{R}^*$.*

Proof: Assume that $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*$ and fix any $a \in [0, R_{\mathsf{c}}]$. The case where $R_{\mathsf{c}} = 0$ is trivial. Next, we consider the case where $R_{\mathsf{c}} > 0$. Recall that $R_{\mathsf{u}}^* = H(f(X,Y)|Y)$. Then, time sharing between $(0, R_{\mathsf{u}}^*)$ and $(R_{\mathsf{c}}, R_{\mathsf{u}})$ asserts that

$$\left(R_{\mathsf{c}} - a, R_{\mathsf{u}} + \frac{R_{\mathsf{u}}^* - R_{\mathsf{u}}}{R_{\mathsf{c}}} a\right) \in \mathcal{R}^*.$$

Then, Corollary 3.1 implies that

$$\frac{R_{\mathsf{u}}^* - R_{\mathsf{u}}}{R_{\mathsf{c}}} \leq 1,$$

so it holds that $(R_{\mathsf{c}} - a, R_{\mathsf{u}} + a) \in \mathcal{R}^*$. ■

Corollary 3.3. *Let $(R_{\mathsf{c}}, R_{\mathsf{u}})$ be an extreme point of $\mathcal{R}^*$ such that $R_{\mathsf{c}} > 0$. Then, for all $a > 0$, $(R_{\mathsf{c}} + a, R_{\mathsf{u}} - a) \notin \mathcal{R}^*$.*

Proof: Recall that $R_{\mathsf{u}}^* = H(f(X,Y)|Y)$. Assume that $(R_{\mathsf{c}} + a, R_{\mathsf{u}} - a) \in \mathcal{R}^*$. Then, time sharing between $(0, R_{\mathsf{u}}^*)$ and $(R_{\mathsf{c}} + a, R_{\mathsf{u}} - a)$ asserts that

$$\left(R_{\mathsf{c}}, \frac{aR_{\mathsf{u}}^* + R_{\mathsf{c}}(R_{\mathsf{u}} - a)}{R_{\mathsf{c}} + a}\right) \in \mathcal{R}^*.$$

However, Corollary 3.1 implies that

$$\frac{aR_{\mathsf{u}}^* + R_{\mathsf{c}}(R_{\mathsf{u}} - a)}{R_{\mathsf{c}} + a} \leq R_{\mathsf{u}},$$

which contradicts the assumption that $(R_{\mathsf{c}}, R_{\mathsf{u}})$ is an extreme point. ■

In general, the optimal sum rate can only be achieved at the point $(R_{\mathsf{c}}, R_{\mathsf{u}}) = (0, H(f(X,Y)|Y))$, i.e., the update encoder does all the work. Nevertheless, for the class of partially invertible functions, one can arbitrarily distribute the work load without compromising the sum rate.

Corollary 3.4. *If the function f is partially invertible, i.e., $H(X|f(X,Y),Y) = 0$, then $R_{\mathsf{c}}^* = R_{\mathsf{u}}^* = H(X|Y)$.*

Proof: First, Theorem 3.1 implies that $R_{\mathsf{u}}^* \le R_{\mathsf{c}}^* \le H(X|Y)$. Then, the corollary is an easy consequence of the property of the function f:

$$R_{\mathsf{u}}^* = H(f(X,Y)|Y) = H(f(X,Y),X|Y) = H(X|Y).$$

■

In other words, Corollary 3.4 says that for partially invertible functions, e.g., arithmetic sum and modulo sum, the side information Y at the update encoder is useless in lowering the compression rate and thus in this case the cache encoder is as powerful as the update encoder. More generally, it can be shown that $R_{\mathsf{c}}^* = R_{\mathsf{u}}^*$ if and only if there exists a conditional pmf $p_{V|X}$ such that

1. $H(V|f(X,Y),Y) = H(V|X,Y)$, and
2. $H(f(X,Y)|V,Y) = 0$.

For most of the problems, it is challenging to find a closed-form expression for the optimal rate region $\mathcal{R}^*$. That is, we do not know the optimal caching strategy in general.[2] In the following, we consider three cases where X and Y are independent, which implies that $I(X;V|Y) = I(X;V)$. For the first two cases, we are able to show that some intuitive caching strategies are indeed optimal. In the last case, we provide some guidance for the optimal caching strategy. Without loss of generality, we assume that $\mathcal{Y} = [N]$. Besides, we will find it convenient to denote $x^{(y)} := f(x,y)$.

Remark 3.1. *If X and Y are independent, then the optimal conditional pmf $p_{V|X}^*$ for a fixed cache rate R_{c} can be found by solving the following constrained optimization problem*

$$\begin{aligned} &\textit{maximize} && I(f(X,Y),Y;V) \\ &\textit{subject to} && I(X;V) \le R_{\mathsf{c}} \end{aligned}$$

over all conditional pmf $p_{V|X}$ with $|\mathcal{V}| \le |\mathcal{X}| + 1$. Thus, caching has an information bottleneck *interpretation [39] (see also [40]). Given a fixed-size cache as bottleneck, we aim to provide the most relevant information of the desired function $f(X,Y)$. We can apply the existing algorithms developed for the information bottleneck method to numerically characterize the optimal rate region $\mathcal{R}^*$.*

[2] A caching strategy is said to be (Pareto) optimal if its achievable rate pair lies on the boundary of the optimal rate region $\mathcal{R}^*$.

3.3 Independent Source Components

In this section, we consider the case where $H(X^{(1)}, \cdots, X^{(N)}) = \sum_{n=1}^{N} H(X^{(n)})$, i.e., the source components are independent of each other. Note that we used the shorthand notation $X^{(n)} = f(X, n)$. Without loss of generality, we assume that $p_Y(1) \geq p_Y(2) \geq \cdots \geq p_Y(N)$. Then, we have the following proposition.

Proposition 3.1. *If X and Y are independent and $X^{(1)}, \cdots, X^{(N)}$ are independent as well, then the optimal rate region $\mathcal{R}^*$ is the set of rate pairs $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that*

$$\begin{aligned} &R_{\mathsf{c}} \geq r, \\ &R_{\mathsf{u}} \geq \sum_{n=1}^{N} (p_Y(n) - p_Y(n+1)) \left(\sum_{j=1}^{n} H(X^{(j)}) - r \right)^+, \end{aligned} \tag{3.4}$$

for some $r \geq 0$, where $p_Y(N+1) = 0$.

Proof: (Converse.) Suppose that $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*$. Then, there exists a conditional pmf $p_{V|X}$ such that $R_{\mathsf{c}} \geq I(X; V|Y) =: r$ and $R_{\mathsf{u}} \geq H(f(X,Y)|V, Y)$. For all $n \in [N]$, we have

$$\begin{aligned} R_{\mathsf{c}} \geq r &= I(X; V|Y) \\ &\overset{(a)}{=} I(X; V) \\ &\geq I(X^{(1)}, \cdots, X^{(n)}; V) \\ &\geq \sum_{j=1}^{n} H(X^{(j)}) - \sum_{j=1}^{n} H(X^{(j)}|V), \end{aligned} \tag{3.5}$$

where (a) follows since X and Y are independent. Next we show that R_{u} can be lower bounded as in (3.4):

$$\begin{aligned} R_{\mathsf{u}} &\geq H(f(X,Y)|V, Y) \\ &= \sum_{j=1}^{N} p_Y(j) H(X^{(j)}|V) \\ &\overset{(a)}{\geq} p_Y(N) \left(\sum_{j=1}^{N} H(X^{(j)}) - r - \sum_{j=1}^{N-1} H(X^{(j)}|V) \right)^+ + \sum_{j=1}^{N-1} p_Y(j) H(X^{(j)}|V) \\ &\overset{(b)}{\geq} p_Y(N) \left(\sum_{j=1}^{N} H(X^{(j)}) - r \right)^+ + \sum_{j=1}^{N-1} (p_Y(j) - p_Y(N)) H(X^{(j)}|V), \end{aligned} \tag{3.6}$$

where (a) follows from (3.5) with $n = N$ and $H(X^{(N)}|V) \geq 0$ and (b) follows since $(u - v)^+ \geq (u)^+ - v$ for all $v \geq 0$. The term on the right-hand side of (3.6) can be

lower bounded as

$$\begin{aligned}
&\sum_{j=1}^{N-1}(p_Y(j)-p_Y(N))H(X^{(j)}|V)\\
&\overset{(a)}{\geq}(p_Y(N-1)-p_Y(N))\left(\sum_{j=1}^{N-1}H(X^{(j)})-r-\sum_{j=1}^{N-2}H(X^{(j)}|V)\right)^+\\
&\quad+\sum_{j=1}^{N-2}(p_Y(j)-p_Y(N))H(X^{(j)}|V)\\
&\geq(p_Y(N-1)-p_Y(N))\left(\sum_{j=1}^{N-1}H(X^{(j)})-r\right)^+\\
&\quad+\sum_{j=1}^{N-2}(p_Y(j)-p_Y(N-1))H(X^{(j)}|V),
\end{aligned}$$

where (a) follows from (3.5) with $n=N-1$ and $H(X^{(N-1)}|V)\geq 0$. At this point, it is clear that we can apply the same argument for another $N-2$ times and arrive at

$$R_{\mathsf{u}}\geq\sum_{n=1}^{N}(p_Y(n)-p_Y(n+1))\left(\sum_{j=1}^{n}H(X^{(j)})-r\right)^+, \tag{3.7}$$

where $p_Y(N+1)=0$.

(Achievability.) Note that the lower bound (3.7) is equivalent to saying that

1. if $r\geq\sum_{n=1}^{N}H(X^{(n)})$, then $R_{\mathsf{u}}\geq 0$, and
2. if $\sum_{j=1}^{n-1}H(X^{(j)})\leq r<\sum_{j=1}^{n}H(X^{(j)})$ for some $n\in[N]$, then

$$R_{\mathsf{u}}\geq p_Y(n)\left(\sum_{j=1}^{n}H(X^{(j)})-r\right)+\sum_{j=n+1}^{N}p_Y(j)H(X^{(j)}).$$

Therefore, for all $n\in\{0\}\cup[N]$, substituting $V=(X^{(1)},\cdots,X^{(n)})$ in (3.2) and (3.3) shows that the rate pair

$$(R_{\mathsf{c}},R_{\mathsf{u}})=\left(\sum_{j=1}^{n}H(X^{(j)}),\sum_{j=n+1}^{N}p_Y(j)H(X^{(j)})\right)$$

is achievable, which corresponds to an extreme point in the region described by $R_{\mathsf{c}}\geq r$ and (3.7). Since the rest of points on the boundary can be achieved by time sharing, the achievability is established. ■

Thus, Proposition 3.1 indicates that the best caching strategy for independent source components is to cache the most popular ones, no matter how different the component sizes are (see also [27] and the references therein).

3.4 Nested Source Components

Again using the shorthand notation $X^{(n)} = f(X, n)$, in this section we assume that $H(X^{(n)}|X^{(n+1)}) = 0$ for all $n \in [N-1]$, i.e., each component corresponds to a refined version of it predecessor, but p_Y can be arbitrary. Then, we have the following proposition.

Proposition 3.2. *If X and Y are independent and $H(X^{(n)}|X^{(n+1)}) = 0$ for all $n \in [N-1]$, then the optimal rate region $\mathcal{R}^*$ is the set of rate pairs $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that*

$$\begin{aligned} R_{\mathsf{c}} &\geq r, \\ R_{\mathsf{u}} &\geq \sum_{n=1}^{N} p_Y(n) \left(H(X^{(n)}) - r \right)^+, \end{aligned} \tag{3.8}$$

for some $r \geq 0$.

Proof: (Converse.) Suppose that $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*$. Then, there exists a conditional pmf $p_{V|X}$ such that $R_{\mathsf{c}} \geq I(X;V|Y) =: r$ and $R_{\mathsf{u}} \geq H(f(X,Y)|V,Y)$. For all $n \in [N]$, we have

$$\begin{aligned} R_{\mathsf{c}} \geq r &= I(X;V|Y) \\ &\overset{(a)}{=} I(X;V) \\ &\geq I(X^{(1)}, \cdots, X^{(n)}; V) \\ &\overset{(b)}{=} H(X^{(n)}) - \sum_{j=1}^{n} H(X^{(j)}|V, X^{(j-1)}), \end{aligned} \tag{3.9}$$

where (a) follows since X and Y are independent and (b) follows from the assumption that $H(X^{(n)}|X^{(n+1)}) = 0$ for all $n \in [N-1]$. Next, we show that R_{u} can be lower bounded as in (3.8):

$$\begin{aligned} R_{\mathsf{u}} &\geq H(f(X,Y)|V,Y) \\ &= \sum_{n=1}^{N} p_Y(n) H(X^{(n)}|V) \\ &\overset{(a)}{=} \sum_{n=1}^{N} p_Y(n) H(X^{(1)}, \cdots, X^{(n)}|V) \\ &\overset{(b)}{=} \sum_{n=1}^{N} p_Y(n) \sum_{j=1}^{n} H(X^{(j)}|V, X^{(j-1)}) \\ &= \sum_{j=1}^{N} \left(\sum_{n=j}^{N} p_Y(n) \right) H(X^{(j)}|V, X^{(j-1)}), \end{aligned}$$

where (a) and (b) follow from the assumption that $H(X^{(n)}|X^{(n+1)}) = 0$ for all $n \in [N-1]$. For notational convenience, let us denote $s_j = \sum_{n=j}^{N} p_Y(n)$ and

$q_j = H(X^{(j)}|V, X^{(j-1)})$. Then, we have

$$\begin{aligned}
R_{\mathsf{u}} &\geq \sum_{j=1}^{N} s_j q_j \\
&\overset{(a)}{\geq} s_N \left(H(X^{(N)}) - r - \sum_{j=1}^{N-1} q_j \right)^+ + \sum_{j=1}^{N-1} s_j q_j \\
&\overset{(b)}{\geq} s_N \left(H(X^{(N)}) - r \right)^+ + \sum_{j=1}^{N-1} (s_j - s_N) q_j \\
&= p_Y(N) \left(H(X^{(N)}) - r \right)^+ + \sum_{j=1}^{N-1} (s_j - s_N) q_j \\
&\overset{(c)}{\geq} p_Y(N) \left(H(X^{(N)}) - r \right)^+ + (s_{N-1} - s_N) \left(H(X^{(N-1)}) - r - \sum_{j=1}^{N-2} q_j \right)^+ \\
&\quad + \sum_{j=1}^{N-2} (s_j - s_N) q_j \\
&\overset{(d)}{\geq} p_Y(N) \left(H(X^{(N)}) - r \right)^+ + (s_{N-1} - s_N) \left(H(X^{(N-1)}) - r \right)^+ \\
&\quad + \sum_{j=1}^{N-2} (s_j - s_{N-1}) q_j \\
&= \sum_{n=N-1}^{N} p_Y(n) \left(H(X^{(n)}) - r \right)^+ + \sum_{j=1}^{N-2} (s_j - s_{N-1}) q_j,
\end{aligned}$$

where (a) and (c) follow from (3.9) and $H(X^{(n)}|V, X^{(n-1)}) \geq 0$ with $n = N$ and $n = N-1$, respectively, and (b) and (d) follow since $(u-v)^+ \geq (u)^+ - v$ for all $v \geq 0$. At this point, it is clear that we can apply the same argument for another $N-2$ times and arrive at

$$R_{\mathsf{u}} \geq \sum_{n=1}^{N} p_Y(n) \left(H(X^{(n)}) - r \right)^+. \tag{3.10}$$

(Achievability.) Note that the lower bound (3.10) is equivalent to saying that

1. if $r \geq H(X^{(N)})$, then $R_{\mathsf{u}} \geq 0$, and
2. if $H(X^{(j-1)}) \leq r < H(X^{(j)})$ for some $j \in [N]$, where $H(X^{(0)}) := 0$, then

$$R_{\mathsf{u}} \geq \sum_{n=j}^{N} p_Y(n) \left(H(X^{(n)}) - r \right)^+.$$

Therefore, for all $n \in [N]$, substituting $V = X^{(n)}$ in (3.2) and (3.3) shows that the rate pair

$$(R_{\mathsf{c}}, R_{\mathsf{u}}) = \left(H(X^{(n)}), \sum_{j=n+1}^{N} p_Y(j) H(X^{(j)}|X^{(n)}) \right),$$

is achievable, which corresponds to an extreme point in the region described by $R_{\mathsf{c}} \geq r$ and (3.10). Since the rest of points on the boundary can be achieved by time sharing, the proposition is established. ■

Thus, Proposition 2 indicates that the best caching strategy for nested components is to cache the coarsest versions up to the cache size.

3.5 Arbitrarily Correlated Components with Uniform Requests

Here we assume that the request is uniformly distributed, i.e., $p_Y(n) = \frac{1}{N}$ for all $n \in [N]$, but $X^{(1)}, \cdots, X^{(N)}$ can be arbitrarily correlated. Recall that $X^{(n)} = f(X, n)$. Although we cannot give a closed-form expression of the optimal rate region, we provide a necessary and sufficient condition on the auxiliary random variable which characterizes the boundary of the optimal rate region.

Proposition 3.3. *If X and Y are independent and $p_Y(n) = \frac{1}{N}$ for all $n \in [N]$, then all points $(R_{\mathsf{c}}, R_{\mathsf{u}})$ on the boundary of the optimal rate region $\mathcal{R}^*$ can be expressed as*

$$\begin{aligned} R_{\mathsf{c}} &= r, \\ R_{\mathsf{u}} &= \frac{1}{N}\left(H(\overline{X}) - r + \min_{\substack{p_{V|\overline{X}} \\ s.t.\ I(\overline{X};V)=r}} C(\overline{X}|V) \right), \end{aligned}$$

for some $r \in [0, H(\overline{X})]$, where $\overline{X} := (X^{(1)}, X^{(2)}, \cdots, X^{(N)})$ and

$$C(\overline{X}|V) := \left[\sum_{n=1}^{N} H(X^{(n)}|V) \right] - H(X^{(1)}, \cdots, X^{(N)}|V).$$

Proof: Denote by $\overline{\mathcal{R}}$ the set of rate pairs $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that

$$\begin{aligned} R_{\mathsf{c}} &\geq I(\overline{X}; V|Y), \\ R_{\mathsf{u}} &\geq H(f(X,Y)|V,Y), \end{aligned}$$

for some conditional pmf $p_{V|X}$. Since $I(X;V|Y) \geq I(\overline{X};V|Y)$, we have $\mathcal{R}^* \subseteq \overline{\mathcal{R}}$. Also, it is easy to see that the rate region $\overline{\mathcal{R}}$ is achievable, so we conclude that $\mathcal{R}^* = \overline{\mathcal{R}}$. By using the assumptions that $I(X;Y) = 0$ and that Y is uniformly distributed, we can simplify the rate expressions as

$$\begin{aligned} R_{\mathsf{c}} &\geq I(\overline{X}; V), \\ R_{\mathsf{u}} &\geq \frac{1}{N} \sum_{n=1}^{N} H(X^{(n)}|V). \end{aligned}$$

Now denote by $p_{V|\overline{X}}$ the conditional pmf induced by the conditional pmf $p_{V|X}$. As can be checked, both $I(\overline{X};V)$ and $\{H(X^{(n)}|V)\}_{n\in[N]}$ can be completely determined by the induced conditional pmf $p_{V|\overline{X}}$. Thus, it suffices to consider the space of all conditional pmfs $p_{V|\overline{X}}$. Finally, noting that

$$\sum_{n=1}^{N} H(X^{(n)}|V) = H(\overline{X}) - I(\overline{X};V) + C(\overline{X}|V),$$

it holds that if $R_{\mathsf{c}} = r \in [0, H(\overline{X})]$, then

$$\min\{R_{\mathsf{u}} : R_{\mathsf{c}} = r, (R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*\} = \frac{1}{N}\left(H(\overline{X}) - r + \min_{\substack{p_{V|\overline{X}} \\ \text{s.t. } I(\overline{X};V)=r}} C(\overline{X}|V)\right).$$

■

If $N = 2$, we have

$$C(\overline{X}|V) = I(X^{(1)}; X^{(2)}|V),$$

so the term $C(X^{(1)}, \cdots, X^{(N)}|V)$ can be interpreted as a generalization of conditional mutual information. In fact, the term $C(X^{(1)}, \cdots, X^{(N)}) = \left[\sum_{n=1}^{N} H(X^{(n)})\right] - H(X^{(1)}, \cdots, X^{(N)})$ was first studied by Watanabe [41] and given the name *total correlation*. Following this convention, we refer to $C(X^{(1)}, \cdots, X^{(N)}|V)$ as *conditional total correlation*. Proposition 3.3 indicates that an optimal caching strategy is to cache a description of the data set that minimizes the conditional total correlation.

When the cache rate is large enough, there exists a conditional pmf $p_{V|X}$ such that the conditional total correlation is zero and thus we have the following corollary.

Corollary 3.5. *The boundary of the region* $\{(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^* : R_{\mathsf{crit}} \le R_{\mathsf{c}} \le H(\overline{X})\}$ *is a segment of the straight line* $R_{\mathsf{c}} + NR_{\mathsf{u}} = H(\overline{X})$*, where*

$$R_{\mathsf{crit}} = \min_{\substack{p_{V|\overline{X}} \\ s.t.\ C(\overline{X}|V)=0}} I(\overline{X}; V).$$

Note that when $N = 2$, R_{crit} is Wyner's common information [16].

Finally, let us consider an example which covers all three mentioned cases.

Example 3.1. *Fix* $q \in [0, \frac{1}{2}]$*. Consider* $Y \sim$ *Uniform*$(\{1, 2\})$ *and* $X = (X^{(1)}, X^{(2)})$*, where* $\langle X^{(1)}, X^{(2)} \rangle$ *is a DSBS(q). Assume that* X *and* Y *are independent. We first consider two extreme cases.*

1. *If* $q = 1/2$*, then the two components are independent and* $\mathcal{R}^* = \{(R_{\mathsf{c}}, R_{\mathsf{u}}) : R_{\mathsf{c}} \ge 0, R_{\mathsf{u}} \ge 0, R_{\mathsf{c}} + 2R_{\mathsf{u}} \ge 2\}$.
2. *If* $q = 0$*, then the two components are nested and* $\mathcal{R}^* = \{(R_{\mathsf{c}}, R_{\mathsf{u}}) : R_{\mathsf{c}} \ge 0, R_{\mathsf{u}} \ge 0, R_{\mathsf{c}} + R_{\mathsf{u}} \ge 1\}$.

Now consider $0 < q < \frac{1}{2}$*. Wyner's common information of* $(X^{(1)}, X^{(2)})$ *is known as [16]*

$$R_{\mathsf{crit}} = 1 + h_{\mathsf{b}}(q) - 2h_{\mathsf{b}}(q'),$$

where $q' = \frac{1}{2}(1 - \sqrt{1 - 2q})$*. Thus, from Corollary 3.5, we have*

$$\min\{R_{\mathsf{u}} : R_{\mathsf{c}} \ge R_{\mathsf{crit}}, (R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*\} = \frac{1}{2}(1 + h_{\mathsf{b}}(q) - R_{\mathsf{c}}).$$

Note that $R_{\mathsf{u}} \ge \frac{1}{2}(1 + h_{\mathsf{b}}(q) - R_{\mathsf{c}})^+$ *is also a valid lower bound for all* $R_{\mathsf{c}} \ge 0$*. Besides, from Corollary 3.1 we have* $R_{\mathsf{c}} + R_{\mathsf{u}} \ge 1$*.*

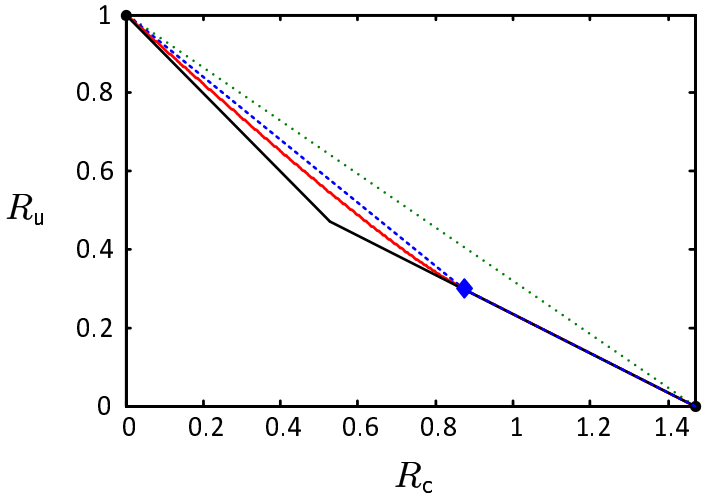

Figure 3.2: Inner bounds and an outer bound for Example 3.1. Here $q = 0.1$.

As for the case $0 < q < \frac{1}{2}$ *and* $R_c < R_{crit}$*, we do not have a complete characterization. Let us consider the following choice of the auxiliary random variable* V*. We set*

$$V = \begin{cases} X^{(1)} \oplus U & \text{if } X^{(1)} = X^{(2)}, \\ W & \text{if } X^{(1)} \neq X^{(2)}, \end{cases} \tag{3.11}$$

where $\oplus$ *denotes modulo-two sum,* $U, W \in \{0,1\}$ *are independent of* (X,Y)*, and furthermore* $W \sim$ *Bernoulli*$(1/2)$*. We conjecture that such choice of* V *characterizes the boundary of* $\mathcal{R}^*$ *for* $R_c < R_{crit}$*. It can be checked that setting*

$$p_U(1) = \frac{1}{2} - \frac{\sqrt{1-2q}}{2(1-q)} =: \gamma$$

achieves Wyner's common information R_{crit}*.*

In Figure 3.2 we plot three inner bounds and an outer bound for the case $q = 0.1$*, where* $R_{crit} \approx 0.873$*. The first inner bound is plotted in green dot, which results from time sharing between the extreme points* $(R_c^*, 0)$ *and* $(0, R_u^*)$*. The extreme point* $\big(R_{crit}, \frac{1}{2}(1 + h_b(q) - R_{crit})\big)$ *is marked by a blue diamond point. Then, the second inner bound is formed by time sharing among the three extreme points. The third inner bound has the same boundary as the second inner bound for* $R_c \geq R_{crit}$*. As for* $R_c < R_{crit}$*, the third inner bound is plotted in red solid, which results from evaluating all* $p_U(1) \in [\gamma, 0.5]$*. Finally, the combined outer bound* $R_u \geq \max\{\frac{1}{2}(1 + h_b(q) - R_c)^+, (1 - R_c)^+\}$ *is plotted in black solid.*

We remark that it can be shown that $H(X^{(n)} \oplus V) = H(X^{(n)}|V)$*,* $n \in \{1,2\}$*. Thus, instead of Slepian–Wolf coding, the update encoder can simply compress* $X^{(n)} \oplus V$ *and transmit. Then, after recovering* $X^{(n)} \oplus V$*, the decoder removes* V *to get the desired component* $X^{(n)}$*.*

3.6 Distributed Compress–Bin with Successive Decoding

The system depicted in Figure 3.1 can be considered as a special case of distributed lossy compression: The cache encoder compresses x^k into a description v^k and the update encoder compresses (x^k, y^k) into another description u^k. Then, the decoder uses the two descriptions (u^k, v^k) with side information y^k to reconstruct an estimate $\{\hat{f}(x_i, y_i)\}_{i\in[k]}$. It can be shown that the distributed compress–bin scheme (see, e.g., [5, Section 12.1]) achieves any rate pair $(R_{\mathsf{c}}, R_{\mathsf{u}})$ satisfying

$$\begin{aligned} R_{\mathsf{c}} &> I(X;V|U,Y), \\ R_{\mathsf{u}} &> I(X;U|V,Y) - I(V;Y), \\ R_{\mathsf{c}} + R_{\mathsf{u}} &> I(X;U,V|Y), \end{aligned}$$

for some conditional pmf $p_{V|X}p_{U|X,Y}$ such that $H(f(X,Y)|U,V,Y) = 0$. Besides, it can be easily checked that the distributed compress–bin scheme achieves all rate pairs in $\mathcal{R}^*$. In the distributed compress–bin scheme, the decoder applies joint decoding on U and V. Now let us consider two achievable rate regions using successive decoding instead. If the description generated by the cache encoder, i.e., V, is recovered first, we refer to the decoding order as "cache $\to$ update". On the other hand, if the description generated by the update encoder, i.e., U, is recovered first, we refer to the decoding order as "update $\to$ cache".

Denote by $\mathcal{R}_{\mathsf{c}\to\mathsf{u}}$ the set of rate pairs $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that

$$\begin{aligned} R_{\mathsf{c}} &> I(X;V|Y), \\ R_{\mathsf{u}} &> I(X;U|V,Y), \end{aligned}$$

for some conditional pmf $p_{V|X}p_{U|X,Y}$ such that $H(f(X,Y)|U,V,Y) = 0$. Also, denote by $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$, the set of rate pairs $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that

$$\begin{aligned} R_{\mathsf{u}} &> I(X;U|Y,Q), \\ R_{\mathsf{c}} &> I(X;V|U,Y,Q), \end{aligned}$$

for some conditional pmf $p_Q p_{V|X,Q}p_{U|X,Y,Q}$ such that $H(f(X,Y)|U,V,Y,Q) = 0$. It can be shown that the rate region $\mathcal{R}_{\mathsf{c}\to\mathsf{u}}$ is achievable by the distributed compress–bin scheme with successive decoding in the order: "cache $\to$ update." If we set $U = f(X,Y)$, then it is easy to see that $\mathcal{R}_{\mathsf{c}\to\mathsf{u}} = \mathcal{R}^*$. Also, it can be shown that the rate region $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$ is achievable by the distributed compress–bin scheme with successive decoding in the order "update $\to$ cache."

However, currently we do not know whether $\mathcal{R}_{\mathsf{u}\to\mathsf{c}} = R^*$ always holds. In the following we provide two conditions on the optimality of successive decoding with the order "update $\to$ cache." The condition given in Proposition 3.4 is a necessary and sufficient condition and the condition given in Proposition 3.5 is a sufficient condition.

Proposition 3.4. *We have $\mathcal{R}_{\mathsf{u}\to\mathsf{c}} = \mathcal{R}^*$ if and only if every extreme point in $\mathcal{R}_{\mathsf{c}\to\mathsf{u}}$ can be described by some conditional pmf $p_{V|X}p_{U|X,Y}$ such that $I(U;V|Y) = 0$.*

Proof: Without loss of generality, we can set $Q = \emptyset$ since we consider only the extreme points. The key observation is that

$$I(U;V|Y) = I(X;V|Y) - I(X;V|U,Y) = I(X;U|Y) - I(X;U|V,Y). \quad (3.12)$$

($\Rightarrow$): It is clear that $(0, R_{\mathsf{u}}^*)$ is an extreme point of both $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$ and $\mathcal{R}_{\mathsf{c}\to\mathsf{u}}$ and can be achieved with $I(U;V|Y) = 0$. Next, consider any extreme point $(R_{\mathsf{c}}, R_{\mathsf{u}})$ of $\mathcal{R}^*$ with $R_{\mathsf{c}} > 0$. Since we assume that $\mathcal{R}_{\mathsf{u}\to\mathsf{c}} = \mathcal{R}^*$, the point $(R_{\mathsf{c}}, R_{\mathsf{u}})$ is also an extreme point of $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$. Therefore, there exists a conditional pmf $p_{V|X}p_{U|X,Y}$ such that $(I(X;V|U,Y), I(X;U|Y)) = (R_{\mathsf{c}}, R_{\mathsf{u}})$. Then, Expression (3.12) implies that $(R_{\mathsf{c}} + I(U;V|Y), R_{\mathsf{u}} - I(U;V|Y)) \in \mathcal{R}_{\mathsf{c}\to\mathsf{u}} = \mathcal{R}^* = \mathcal{R}_{\mathsf{u}\to\mathsf{c}}$. Since $(R_{\mathsf{c}}, R_{\mathsf{u}})$ is an extreme point, Lemma 3.3 implies that $I(U;V|Y) = 0$.

($\Leftarrow$): It suffices to show that every extreme point of $\mathcal{R}^*$ also lies in $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$. Let $(R_{\mathsf{c}}, R_{\mathsf{u}})$ be any extreme point of $\mathcal{R}^*$. From the assumption, there exists a conditional pmf $p_{V|X}p_{U|X,Y}$ achieving $(R_{\mathsf{c}}, R_{\mathsf{u}})$ such that $I(U;V|Y) = 0$. Then, it implies that

$$\begin{aligned}(R_{\mathsf{c}}, R_{\mathsf{u}}) &= (I(X;V|Y), I(X;U|V,Y)) \\ &= (I(X;V|U,Y), I(X;V|Y))\end{aligned}$$

lies in $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$. ∎

Proposition 3.5. *If every extreme point in $\mathcal{R}_{\mathsf{c}\to\mathsf{u}}$ can be described by some conditional pmf $p_{V|X}p_{U|X,Y}$ such that $H(V|X) = 0$, then $\mathcal{R}_{\mathsf{u}\to\mathsf{c}} = \mathcal{R}^*$.*

Proof: It suffices to show that every extreme point in $\mathcal{R}^*$ lies in $\mathcal{R}_{\mathsf{u}\to\mathsf{c}}$. Consider any extreme point $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}^*$. Since $\mathcal{R}^* = \mathcal{R}_{\mathsf{c}\to\mathsf{u}}$, there exists a conditional pmf $p_{V|X}p_{U|X,Y}$ such that

$$\begin{aligned}R_{\mathsf{c}} &= I(X;V|Y), \\ R_{\mathsf{u}} &= I(X;U|V,Y),\end{aligned}$$

with $H(f(X,Y)|U,V,Y) = 0$. Now consider any $y \in \mathcal{Y}$. Conditioned on $Y = y$, the functional representation lemma [5, Appendix B] says that there exists a random variable $U^{(y)}$ of cardinality $|\mathcal{U}^{(y)}| \leq |\mathcal{V}|(|\mathcal{U}| - 1) + 1$ such that

1. $H(U|U^{(y)}, V, Y = y) = 0$,
2. $I(U^{(y)}; V|Y = y) = 0$,
3. $I(X; U^{(y)}|U, V, Y = y) = 0$.

Therefore, we have $I(U^{(Y)}; V|Y) = 0$ and $I(X;U|V,Y) = I(X;U^{(Y)}|V,Y)$. Together with (3.12), it implies that

$$\begin{aligned}R_{\mathsf{c}} &= I(X;V|U^{(Y)},Y), \\ R_{\mathsf{u}} &= I(X;U^{(Y)}|Y).\end{aligned}$$

Finally, from the assumption $H(V|X) = 0$, we have $p_{U^{(Y)},V|X,Y} = p_{V|X}p_{U^{(Y)}|X,Y}$ and thus $(R_{\mathsf{c}}, R_{\mathsf{u}}) \in \mathcal{R}_{\mathsf{u}\to\mathsf{c}}$. ∎

We remark that if X and Y are independent, then for both the case of independent components and the case of nested components, Proposition 3.5 implies that $\mathcal{R}_{\mathsf{u}\to\mathsf{c}} = \mathcal{R}^*$.

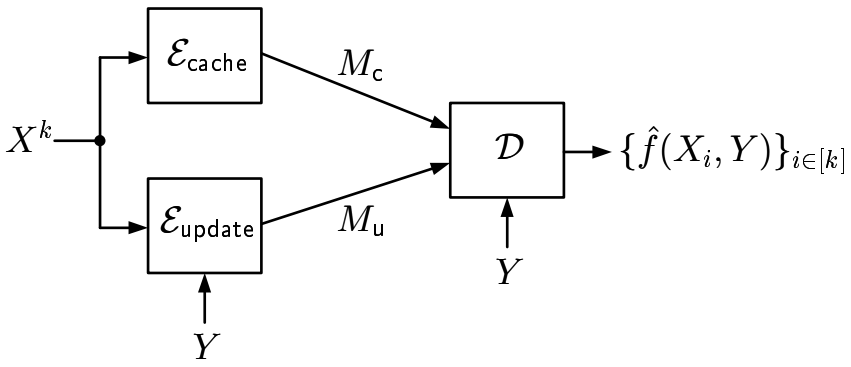

Figure 3.3: The single-request model, in which we assume that X and Y are independent.

3.7 The Single-Request Model

In this section, we discuss the "single-request" model mentioned in the beginning of this chapter. The system is depicted in Figure 3.3. Fix a joint distribution $p_{XY} = p_X p_Y$, i.e., we assume that X and Y are independent. The database is still modeled by a DMS $\langle X \rangle$, which generates an i.i.d. source sequence X^{kB}. In the multi-request model, each X_i is paired with a distinct Y_i, $i \in [kB]$. By contrast, in the single-request model, only each block $(X_{(j-1)B+1}, \cdots, X_{(j-1)B+k})$ is paired with a distinct Y_j, $j \in [B]$, where Y^B are i.i.d. drawn from p_Y. Thus, the request varies less frequently in the single-request model. We can draw an analogy between the request models for content delivery networks and the fading models for wireless networks. If we think of the requests as channel states, then the requests behave like *fast fading* in the multi-request model. On the other hand, in the single-request model, the requests behave like *slow fading*.

Here we assume that each block $j \in [B]$ is processed independently. Then, without loss of generality, we restrict attention to the first block, i.e., $j = 1$. Furthermore, we assume that $\mathcal{Y} = [N]$, where $N \in \mathbb{Z}^+$. The cache encoder observes the source sequence X^k, the update encoder observes the source sequence X^k and the single request Y_1, and the decoder only observes the request Y_1.[3] The cache encoder generates a message $M_{\mathsf{c}} \in [2^{kR_{\mathsf{c}}}]$, and the update encoder generates a message $M_{\mathsf{u}}(Y) \in [2^{kR_{\mathsf{u}}(Y)}]$. The decoder receives $(M_{\mathsf{c}}, M_{\mathsf{u}}(Y))$ and wishes to recover the sequence of functions $\{f(X_i, Y)\}_{i\in[k]}$ losslessly. Note that the length of the update message is a function of Y.

A $(2^{kR_{\mathsf{c}}}, \{2^{kR_{\mathsf{u}}(y)}\}_{y\in[N]}, k)$ distributed multiple description code consists of

- one cache encoder, which assigns an index $m_{\mathsf{c}}(x^k) \in [2^{kR_{\mathsf{c}}}]$ to each sequence $x^k \in \mathcal{X}^k$;
- one update encoder, which assigns N indices $m_{\mathsf{u}}(x^k, y) \in [2^{kR_{\mathsf{u}}(y)}]$, $y \in [N]$, to each sequence $x^k \in \mathcal{X}^k$;
- one decoder, which assigns an estimate $\hat{s}^k$ to each tuple $(m_{\mathsf{c}}, m_{\mathsf{u}}, y)$.

[3] For notational convenience, we drop the subscript 1 in Y_1 hereafter.

In words, the update encoder and the decoder each has N codebooks, each of which is served for one request $y \in [N]$. On the other hand, since the cache encoder does not observe the request Y and cannot infer anything about Y from the observed source sequence X^k, the cache encoder only needs one codebook.

In practice, the set of requested functions has to be agreed by the server and the user in advance. Later on, if the user sends a request not in $[N]$, then the server can refuse to act upon such request. Otherwise, the server must deliver the agreed service. Therefore, we assume that the recovered sequence of estimates $\hat{S}^k$ must satisfy that

$$\lim_{k\to\infty} \mathbb{P}\left(\bigcup_{i\in[k]} \left\{ \hat{S}_i \neq f(X_i, y) \right\} \right) = 0, \quad \forall y \in [N]. \tag{3.13}$$

Thus, for the single-request model, we say that a rate tuple $(R_{\mathsf{c}}, \{R_{\mathsf{u}}(y)\}_{y\in[N]})$ is achievable if there exists a sequence of $(2^{kR_{\mathsf{c}}}, \{2^{kR_{\mathsf{u}}(y)}\}_{y\in[N]}, k)$ codes such that (3.13) holds. Similarly, the optimal rate region $\mathcal{R}^*$ is the closure of the set of achievable rate pairs.

The above problem can be considered as an N-user Gray–Wyner system: Each user $y \in [N]$ receives a private message $M_{\mathsf{u}}(y)$ and a common message M_{c}. The common message is received by all users. Each user $y \in [N]$ wishes to recover the sequence $\{f(X_i, y)\}_{i\in[k]}$. Then, we have the following theorem.

Theorem 3.2. *Consider the single-request model. The optimal rate region $\mathcal{R}^*$ is the set of rate tuples $(R_{\mathsf{c}}, \{R_{\mathsf{u}}(y)\}_{y\in[N]})$ such that*

$$\begin{aligned} R_{\mathsf{c}} &\geq I(X; V), \\ R_{\mathsf{u}}(y) &\geq H(f(X, y)|V), \quad \forall y \in [N], \end{aligned}$$

for some conditional pmf $p_{V|X}$ with $|\mathcal{V}| \leq |\mathcal{X}| + N$.

The single-request model can be studied under more specific performance criteria, which relate the request-dependent update rates $\{R_{\mathsf{u}}(y)\}$ to a fixed quantity R_{u}, which is independent of the realization of Y. In the following, we discuss three common performance criteria by analogy with slow fading. For convenience, we denote

$$\mathcal{R}^*(R_{\mathsf{c}}) = \{\mathbf{R} : (r, \mathbf{R}) \in \mathcal{R}^*, r \leq R_{\mathsf{c}}\},$$

where $\mathbf{R} = (R_{\mathsf{u}}(1), R_{\mathsf{u}}(2), \cdots, R_{\mathsf{u}}(N))$.

3.7.1 Compound

Consider a fixed update rate R_{u}. The compound formulation requires that for each block of length k, the update message cannot contain more than $\lceil kR_{\mathsf{u}} \rceil$ bits. Namely, it requires that for all $y \in [N]$,

$$R_{\mathsf{u}}(y) \leq R_{\mathsf{u}}.$$

Thus, the compound-optimal update rate given the cache rate R_{c} can be defined as

$$R_{\mathsf{compound}}(R_{\mathsf{c}}) := \min_{\mathbf{R}\in\mathcal{R}^*(R_{\mathsf{c}})} \max_{y\in[N]} R_{\mathsf{u}}(y).$$

Then, from Theorem 3.2, we have the following corollary.

Corollary 3.6. *Consider the compound formulation of the single-request model. The compound-optimal update rate given the cache rate R_{c} can be expressed as*

$$R_{\mathsf{compound}}(R_{\mathsf{c}}) = \min_{\substack{p_{V|X} \\ s.t.\ I(X;V)\leq R_{\mathsf{c}}}} \max_{y\in[N]} H(f(X,y)|V),$$

where $|\mathcal{V}| \leq |\mathcal{X}| + N$.

The compound formulation aims to model the worst-case scenario in which the request statistics is not known and/or the communication resource cannot be redistributed over blocks. However, for most applications, the compound formulation is too pessimistic.

3.7.2 Outage

The criterion is similar to the compound formulation except that now an excess probability ρ is permitted. With probability ρ, the server has to assign extra communication resource to fulfill the user's wish. We say that an *outage* occurs if the required update rate $R_{\mathsf{u}}(Y)$ is larger than the preallocated update rate R_{u}. Then, the outage-optimal update rate given the cache rate R_{c} can be defined as

$$R_{\mathsf{outage}}(R_{\mathsf{c}}) := \min_{\mathbf{R}\in\mathcal{R}^*(R_{\mathsf{c}})} \inf\{r : \mathbb{P}(R_{\mathsf{u}}(Y) > r) \leq \rho\}.$$

Similarly, from Theorem 3.2 we have the following corollary.

Corollary 3.7. *Consider the outage formulation of the single-request model. The outage-optimal update rate given the cache rate R_{c} and the excess probability ρ can be expressed as*

$$R_{\mathsf{outage}}(R_{\mathsf{c}},\rho) = \min_{\substack{p_{V|X} \\ s.t.\ I(X;V)\leq R_{\mathsf{c}}}} \inf\{r : \mathbb{P}(\phi(Y) > r) \leq \rho\},$$

where $\phi(y) = H(f(X,y)|V)$, for all $y \in [N]$, and $|\mathcal{V}| \leq |\mathcal{X}| + N$.

Note that $\phi(Y)$ is a random variable induced by the random variable Y.

The outage formulation has the similar issue as the compound formulation. That is, it does not consider the fact that for most applications, the communication resource can indeed be redistributed over blocks. Thus, a model considering the average behavior would be a better fit. Let us discuss such model to end this chapter.

3.7.3 Adaptive Coding

Different from the previous two criteria, here the unused communication resource can be saved for the other blocks. The only requirement is that the average number of bits per block cannot be larger than $\lceil kR_{\mathsf{u}} \rceil$. To see the relation of the adaptive coding formulation with the multi-request model, let us consider the whole rate region instead. We define the adaptive-optimal rate region as

$$\mathcal{R}_{\mathsf{adaptive}} := \left\{(R_{\mathsf{c}}, \mathbb{E}_Y[R_{\mathsf{u}}(Y)]) : (R_{\mathsf{c}}, \{R_{\mathsf{u}}(y)\}_{y\in[N]}) \in \mathcal{R}^*\right\}.$$

From Theorem 3.2 we have the following corollary.

Corollary 3.8. *Consider the adaptive coding formulation of the single-request model. The adaptive-optimal rate region* $\mathcal{R}_{\mathsf{adaptive}}$ *is the set of rate pairs* $(R_{\mathsf{c}}, R_{\mathsf{u}})$ *such that*

$$\begin{aligned} R_{\mathsf{c}} &\geq I(X;V), \\ R_{\mathsf{u}} &\geq H(f(X,Y)|V,Y), \end{aligned}$$

for some conditional pmf $p_{V|X}$ *with* $|\mathcal{V}| \leq |\mathcal{X}| + 1$.

Proof: It suffices to show that $\mathbb{E}_Y[\phi(Y)] = H(f(X,Y)|V,Y)$, where $\phi(y) = H(f(X,y)|V)$, for all $y \in [N]$. Indeed, we have

$$\begin{aligned} E_Y[\phi(Y)] &= \sum_{y=1}^{N} p_Y(y)\phi(y) \\ &= \sum_{y=1}^{N} p_Y(y) H(f(X,y)|V) \\ &\overset{(a)}{=} \sum_{y=1}^{N} p_Y(y) H(f(X,y)|V, Y=y) \\ &= H(f(X,Y)|V,Y), \end{aligned}$$

where (a) follows since X is independent of Y, by assumption, and $V \multimap\!\!- X \multimap\!\!- Y$ form a Markov chain. Finally, we remark that the cardinality bound on $\mathcal{V}$ is refined from $|\mathcal{X}| + N$ to $|\mathcal{X}| + 1$, which can be proved using the convex cover method [5, Appendix C]. ■

As can be seen, assuming X and Y are independent, the adaptive-optimal rate region of the single-request model is the same as the optimal rate region of the multi-request model. Actually, it can be verified that all inner bounds and outer bounds on the subproblems of the multi-request model (see Chapter 4) also work for the corresponding subproblems of the single-request model under the adaptive coding formulation.

Sequential Coding for Computing – The Two-User Case

4

In this chapter, we move on to the two-user case of the problem of sequential coding for computing.[1] Recall that we refer to the first stage as cache and the second stage as update. One new feature for the two-user case is that we can deploy a *common cache* that is shared by both users. When both users have common interests, the content stored in the common cache can be useful for both of them. Similarly, we can deploy a common link from the update encoder to the end users so that both of them can receive a *common update*. Therefore, for the two-user case, there are totally six messages: two private caches, two private updates, one common cache, and one common update.

Since end users usually have distinct requests, we use the notation $\underline{Y}_\ell$ to denote the request of User ℓ. Furthermore, each end user may have his/her own request patterns, so we use the notation g_ℓ. For most applications, requests require much less communication resources compared to the requested data. For simplicity, we assume that the update encoder first broadcasts the requests to both users. That is, denoting by $\underline{Y} = (\underline{Y}_1, \underline{Y}_2)$ the collection of requests, we assume that $\underline{Y}$ is globally known except to the cache encoder. Thus, we denote $f_\ell(\underline{X}, \underline{Y}) = g_\ell(\underline{X}, \underline{Y}_\ell)$ and focus on the functions $\{f_\ell(\cdot,\cdot)\}$ hereafter. As in Chapter 3, we restrict attention to the average-case analysis of the multi-request model.

Unfortunately, even with such simplification, a complete single-letter characterization involving all six messages is unavailable at this point. Rather than directly showing the most general achievability involving every message, here we find it more informative to take a *bottom-up* approach. That is, after providing a formal problem statement in Section 4.1, we consider five representative subproblems (summarized in Figure 4.1) first:

1. one common cache, two private caches, and two private updates (Figure 4.2);

2. one common cache, one common update, and two private updates (Figure 4.3);

[1]The material of this chapter has appeared in
1. C.-Y. Wang, S. H. Lim, and M. Gastpar, "Information-theoretic caching," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Hong Kong, China, Jun. 2015.
2. C.-Y. Wang, S. H. Lim, and M. Gastpar, "Information-theoretic caching: Sequential coding for computing," in *arXiv:1504.00553[cs.IT]*, Apr. 2015.

		Configuration				
		1	2	3	4	5
Cache	Common	✓	✓	✓		
	Private 1	✓			✓	✓
	Private 2	✓		✓		✓
Update	Common		✓	✓	✓	✓
	Private 1	✓	✓			
	Private 2	✓	✓	✓	✓	

Figure 4.1: The five representative configurations.

3. one common cache, one common update, one private cache for User 2, and one private update for User 2 (Figure 4.4);

4. one private cache for User 1, one common update, and one private update for User 2 (Figure 4.5);

5. two private caches and one common update (Figure 4.6).

The first three subproblems correspond to the subproblems with the largest number of messages for which we are able to characterize the optimal rate regions. Subproblems 1 and 2 are extensions of the Gray–Wyner system and Subproblem 3 is a special case of distributed successive refinement, which will be discussed later in Chapter 5. The rest two subproblems are the subproblems with the smallest number of messages whose optimal rate regions remains unknown. In the end of the chapter, we will provide a general achievable scheme built upon the developed achievabilities for different subproblems.

To provide some interesting insights, we find it convenient to classify our coding strategies via a concept that we will refer to as "decoding order." More precisely, in all achievable schemes considered in this chapter, the decoders will proceed in multiple steps according to a successive decoding logic: A first description of the source is recovered and then used as side information in the recovery of the second description, and so on. As discussed in Section 3.6 for the single-user case, employing the decoding order "cache $\to$ update" is without loss of optimality. Recall that for the decoding order "cache $\to$ update," the decoder first recovers the description sent by the cache encoder, and then recovers the description sent by the update encoder. A second example is the Gray and Wyner system reviewed in Section 2.3. For this scenario, one of the two successive decoding schemes is that each decoder first recovers the common description, and then leverages the private description in order to fully decode the object of interest. Here, we would thus say that the decoding order is "common $\to$ private." Again, for this special case, Theorem 2.4

shows that employing this decoding order is without loss of optimality. In general, how should we prioritize the decoding order among private cache, common cache, private update, and common update?

4.1 Problem Statement

The two-user case of the problem of sequential coding for computing is formulated as follows. A DMS $\langle X, Y\rangle$ generates i.i.d. source sequences (X^k, Y^k). There are two encoding terminals and two decoding terminals. The *cache* encoder observes the source sequence X^k, the *update* encoder observes the source sequences (X^k, Y^k), and both decoders observe Y^k. Decoder $\ell \in \{1,2\}$ wishes to recover an element-wise function $f_\ell(x,y)$ losslessly. The cache encoder generates three messages $M_{\mathsf{c},\{1\}}$, $M_{\mathsf{c},\{2\}}$, and $M_{\mathsf{c},\{1,2\}}$ of rate $R_{\mathsf{c},\{1\}}$, $R_{\mathsf{c},\{2\}}$, and $R_{\mathsf{c},\{1,2\}}$, respectively. Similarly, the update encoder generates three messages $M_{\mathsf{u},\{1\}}$, $M_{\mathsf{u},\{2\}}$, and $M_{\mathsf{u},\{1,2\}}$ of rate $R_{\mathsf{u},\{1\}}$, $R_{\mathsf{u},\{2\}}$, and $R_{\mathsf{u},\{1,2\}}$, respectively. We note that some of these rates may be zero. Then, Decoder $\ell \in \{1,2\}$ receives the set of messages $(M_{\mathsf{c},\{\ell\}}, M_{\mathsf{c},\{1,2\}}, M_{\mathsf{u},\{\ell\}}, M_{\mathsf{u},\{1,2\}})$. The messages $M_{\mathsf{c},\{\ell\}}$ and $M_{\mathsf{u},\{\ell\}}$ are called *private cache* and *private update*, respectively. Besides, the messages $M_{\mathsf{c},\{1,2\}}$ and $M_{\mathsf{u},\{1,2\}}$ are called *common cache* and *common update*, respectively.

Denote $n_{\mathsf{a}} = 2^{kR_{\mathsf{a}}}$ for any subscript a. Also, for convenience we denote

$$\begin{aligned}\mathbf{n} &:= (n_{\mathsf{c},\{1\}}, n_{\mathsf{c},\{2\}}, n_{\mathsf{c},\{1,2\}}, n_{\mathsf{u},\{1\}}, n_{\mathsf{u},\{2\}}, n_{\mathsf{u},\{1,2\}}),\\ \mathbf{R} &:= (R_{\mathsf{c},\{1\}}, R_{\mathsf{c},\{2\}}, R_{\mathsf{c},\{1,2\}}, R_{\mathsf{u},\{1\}}, R_{\mathsf{u},\{2\}}, R_{\mathsf{u},\{1,2\}}).\end{aligned}$$

An $(\mathbf{n}, k)$ distributed multiple description code consists of $(\mathcal{A} \in \{\{1\},\{2\},\{1,2\}\})$

- two encoders, where the cache encoder assigns three indices $m_{\mathsf{c},\mathcal{A}}(x^k) \in [n_{\mathsf{c},\mathcal{A}}]$ to each sequence $x^k \in \mathcal{X}^k$ and the update encoder assigns three indices $m_{\mathsf{u},\mathcal{A}}(x^k, y^k) \in [n_{\mathsf{u},\mathcal{A}}]$ to each pair of sequences $(x^k, y^k) \in \mathcal{X}^k \times \mathcal{Y}^k$;
- two decoders, where Decoder $\ell \in \{1,2\}$ assigns an estimate $\hat{s}_\ell^k$ to each tuple $(m_{\mathsf{c},\{\ell\}}, m_{\mathsf{c},\{1,2\}}, m_{\mathsf{u},\{\ell\}}, m_{\mathsf{u},\{1,2\}}, y^k)$.

A rate tuple $\mathbf{R}$ is said to be achievable if there exists a sequence of $(\mathbf{n}, k)$ codes with

$$\lim_{k\to\infty} \mathbb{P}\left(\bigcup_{\ell\in\{1,2\}} \bigcup_{i\in[k]} \left\{ \hat{S}_{\ell,i} \neq f_\ell(X_i, Y_i) \right\} \right) = 0.$$

The optimal rate region $\mathcal{R}^*$ is the closure of the set of achievable rate tuples.

We are also interested in the subsets of $\mathcal{R}^*$, in which some of the rate components are set to zero. Let $\mathcal{A}_{\mathsf{c}}$ and $\mathcal{A}_{\mathsf{u}}$ be any subsets of $\{\{1\},\{2\},\{1,2\}\}$. We use the notation $(\mathcal{A}_{\mathsf{c}}|\mathcal{A}_{\mathsf{u}})$, termed *configuration*, to specify the available caches and updates in the system. For example, Configuration $(\mathcal{A}_{\mathsf{c}}|\mathcal{A}_{\mathsf{u}}) = (\{1\},\{2\}|\{1,2\})$ says that each user has his/her own private cache and there is an update common to both users. Then, we define

$$\mathcal{R}^*(\mathcal{A}_{\mathsf{c}}|\mathcal{A}_{\mathsf{u}}) := \{\mathbf{R} \in \mathcal{R}^* : R_{\mathsf{c},\mathcal{B}_{\mathsf{c}}} = 0, R_{\mathsf{u},\mathcal{B}_{\mathsf{u}}} = 0 \text{ for all } \mathcal{B}_{\mathsf{c}} \notin \mathcal{A}_{\mathsf{c}}, \mathcal{B}_{\mathsf{u}} \notin \mathcal{A}_{\mathsf{u}}\}. \quad (4.1)$$

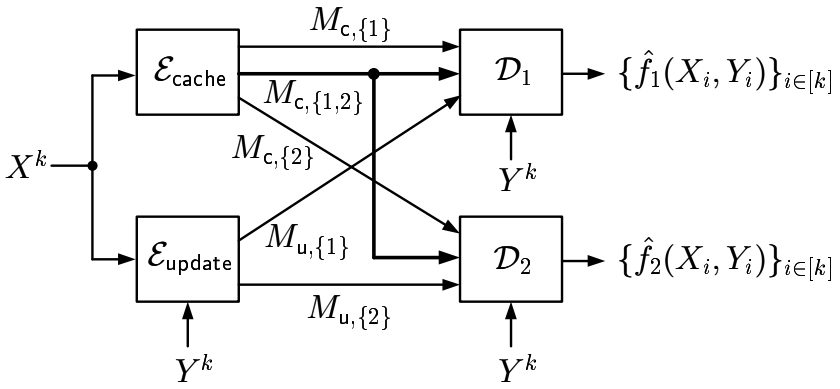

Figure 4.2: The system with Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{1\},\{2\},\{1,2\}|\{1\},\{2\})$.

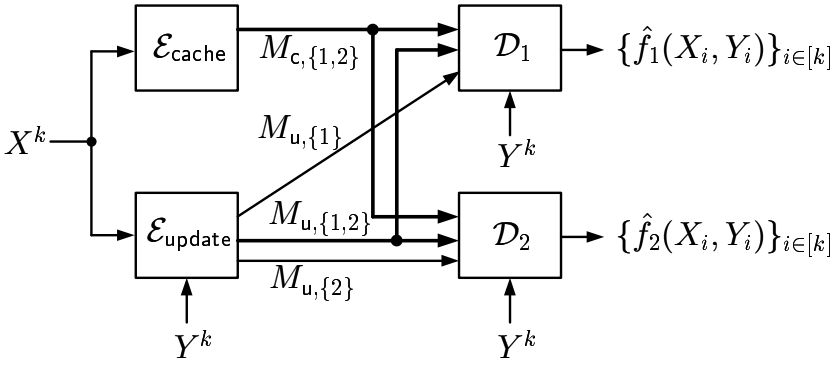

Figure 4.3: The system with Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{1,2\}|\{1\},\{2\},\{1,2\})$.

4.2 Extension of the Gray–Wyner System

We first consider Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{1\},\{2\},\{1,2\}|\{1\},\{2\})$ and Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{1,2\}|\{1\},\{2\},\{1,2\})$, depicted in Figures 4.2 and 4.3, respectively. These configurations include the Gray–Wyner system [9] as a special case. To see this, in Configuration $(\{1\},\{2\},\{1,2\}|\{1\},\{2\})$ set $R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$, and in Configuration $(\{1,2\}|\{1\},\{2\},\{1,2\})$ set $R_{\mathsf{c},\{1,2\}} = 0$. We follow the principle "common $\rightarrow$ private" and establish the optimal rate regions of the two configurations in the following theorems. The proofs are omitted since they follow similar lines as the proof of Theorem 2.4, which can be found in [9].

Theorem 4.1. *The rate region* $\mathcal{R}^*(\{1\},\{2\},\{1,2\}|\{1\},\{2\})$ *is the set of rate tuples* $\mathbf{R}$ *such that* $R_{\mathsf{u},\{1,2\}} = 0$*,*

$$\begin{aligned}
R_{\mathsf{c},\{1,2\}} &\geq I(X;V_\mathsf{c}|Y),\\
R_{\mathsf{c},\{1\}} &\geq I(X;V_1|V_\mathsf{c},Y),\\
R_{\mathsf{c},\{2\}} &\geq I(X;V_2|V_\mathsf{c},Y),\\
R_{\mathsf{u},\{1\}} &\geq H(f_1(X,Y)|V_\mathsf{c},V_1,Y),\\
R_{\mathsf{u},\{2\}} &\geq H(f_2(X,Y)|V_\mathsf{c},V_2,Y),
\end{aligned}$$

for some conditional pmf $p_{V_\mathsf{c}|X}p_{V_1|V_\mathsf{c},X}p_{V_2|V_\mathsf{c},X}$ *satisfying* $|\mathcal{V}_\mathsf{c}| \le |\mathcal{X}| + 4$, $|\mathcal{V}_j| \le |\mathcal{V}_\mathsf{c}||\mathcal{X}| + 1$, $j \in \{1, 2\}$.

Theorem 4.2. *The rate region* $\mathcal{R}^*(\{1,2\}|\{1\},\{2\},\{1,2\})$ *is the set of rate tuples* $\mathbf{R}$ *such that* $R_{\mathsf{c},\{1\}} = R_{\mathsf{c},\{2\}} = 0$,

$$\begin{aligned} R_{\mathsf{c},\{1,2\}} &\ge I(X; V_\mathsf{c}|Y), \\ R_{\mathsf{u},\{1,2\}} &\ge I(X; U|V_\mathsf{c}, Y), \\ R_{\mathsf{u},\{1\}} &\ge H(f_1(X,Y)|U, V_\mathsf{c}, Y), \\ R_{\mathsf{u},\{2\}} &\ge H(f_2(X,Y)|U, V_\mathsf{c}, Y), \end{aligned}$$

for some conditional pmf $p_{V_\mathsf{c}|X}p_{U|V_\mathsf{c},X,Y}$ *satisfying* $|\mathcal{V}_\mathsf{c}| \le |\mathcal{X}|+3$, $|\mathcal{U}| \le |\mathcal{V}_\mathsf{c}||\mathcal{X}||\mathcal{Y}|+2$.

As in the Gray–Wyner system, the optimal decoding order is to first recover the common descriptions at both decoders first, and then each decoder recovers their respective private descriptions. Moreover, in Configuration $(\{1\},\{2\},\{1,2\}|\{1\},\{2\})$, to account for the additional update encoder with private links connected to the decoders, the private descriptions are decoded successively with the order "cache → update". Similarly, to account for the additional cache encoder that has a common link to both decoders in Configuration $(\{1,2\}|\{1\},\{2\},\{1,2\})$, the common descriptions are decoded successively with the order "cache → update". In summary, the rate expressions in Theorems 4.1 and 4.2 are established with the following optimal decoding orders:

1. common cache → private cache → private update, and
2. common cache → common update → private update.

The same principle can be extended to the general multi-user case. For all configurations such that there is no conflict between the two decoding orders "cache → update" and "common → private", a single-letter characterization of the optimal rate region can be found. For the rest of the chapter, we consider three configurations in which there is a conflict between the two principles "cache → update" and "common → private".

4.3 Sequential Successive Refinement

Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{2\},\{1,2\}|\{2\},\{1,2\})$ (see Figure 4.4) is a special case of the problem of distributed computing with successive refinement, which will be discussed in Chapter 5. In the first stage, the cache encoder and the private encoder each send a coarse description to both decoders, and then in the second stage they each send a refined description only to Decoder 2. Here we can see a conflict between the principles "cache → update" and "common → private". It is not clear in the first place whether Decoder 2 should decode the common update content first or the private cache content first.

As shown in the following theorem, it turns out that the following decoding order is optimal for Decoder 2:

common cache → common update → private cache → private update.

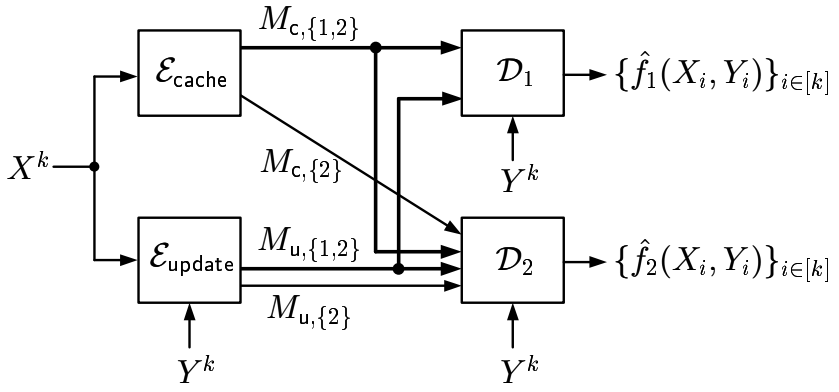

Figure 4.4: The source network with Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{2\},\{1,2\}|\{2\},\{1,2\})$.

Theorem 4.3. *The rate region* $\mathcal{R}^*(\{2\},\{1,2\}|\{2\},\{1,2\})$ *is the set of rate tuples* $\mathbf{R}$ *such that* $R_{\mathsf{c},\{1\}} = R_{\mathsf{u},\{1\}} = 0$,

$$\begin{aligned}
R_{\mathsf{c},\{1,2\}} &\geq I(X;V_\mathsf{c}|Y),\\
R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{2\}} &\geq I(X;V_\mathsf{c}|Y) + I(X;V_2|f_1(X,Y),V_\mathsf{c},Y),\\
R_{\mathsf{u},\{1,2\}} &\geq H(f_1(X,Y)|V_\mathsf{c},Y),\\
R_{\mathsf{u},\{1,2\}} + R_{\mathsf{u},\{2\}} &\geq H(f_1(X,Y)|V_\mathsf{c},Y) + H(f_2(X,Y)|V_2,f_1(X,Y),V_\mathsf{c},Y),
\end{aligned}$$

for some conditional pmf $p_{V_2,V_\mathsf{c}|X}$ *satisfying* $|\mathcal{V}_\mathsf{c}| \leq |\mathcal{X}|+3$ *and* $|\mathcal{V}_2| \leq |\mathcal{V}_\mathsf{c}||\mathcal{X}|+1$.

Proof: (Achievability.) The achievability can be proved by applying Theorem 3.1 and its straightforward extension. Here we provide a high-level description. Consider a simple two-stage coding. In the first stage, we communicate the function f_1 using an optimal multiple description code for the single-user case. Each encoder sends its generated message through its common link. Since both messages $(M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}})$ are also received by Decoder 2, Decoder 2 can also learn $(v_\mathsf{c}^k, \{f_1(x_i,y_i)\}_{i\in[k]})$. Then, in the second stage, we use another multiple description code for the single-user case to communicate the function f_2 but with the augmented side information $(v_\mathsf{c}^k, \{f_1(x_i,y_i)\}_{i\in[k]}, y^k)$. Depending on the rate allocation, each encoder can divide its message of the second stage into two parts, one of which is sent through the common link and the other is sent through the private link.

(Converse.) Denote $S_{1i} = f_1(X_i,Y_i)$ and $S_{2i} = f_2(X_i,Y_i)$ for all $i \in [k]$. The rates $R_{\mathsf{c},\{1,2\}}$ and $R_{\mathsf{u},\{1,2\}}$ can be lower bounded in the same manner as the single-user case and thus the details are omitted. Denote $V_{\mathsf{c}i} = (M_{\mathsf{c},\{1,2\}}, S_1^{i-1}, Y^{[k]\setminus\{i\}})$. Now consider the lower bounds on sum rates $R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{2\}}$ and $R_{\mathsf{u},\{1,2\}} + R_{\mathsf{u},\{2\}}$. First, we have

$$\begin{aligned}
&k(R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{2\}})\\
&\geq H(M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}|Y^k)\\
&= I(X^k; M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}|Y^k)\\
&= I(S_1^k, X^k; M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}|Y^k)\\
&\geq I(S_1^k; M_{\mathsf{c},\{1,2\}}|Y^k) + I(X^k; M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}|S_1^k, Y^k)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{k} I(S_{1i}; M_{\mathsf{c},\{1,2\}} | S_1^{i-1}, Y^k) + \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}} | X^{i-1}, S_1^k, Y^k) \\
&\overset{(a)}{=} \sum_{i=1}^{k} I(S_{1i}; M_{\mathsf{c},\{1,2\}}, S_1^{i-1}, Y^{[k]\setminus\{i\}} | Y_i) \\
&\quad + \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{1,2\}}, S_1^{i-1}, Y^{[k]\setminus\{i\}}, M_{\mathsf{c},\{2\}}, X^{i-1}, S_{1,i+1}^k | S_{1i}, Y_i) \\
&= \sum_{i=1}^{k} I(S_{1i}; V_{\mathsf{c}i} | Y_i) + \sum_{i=1}^{k} I(X_i; V_{\mathsf{c}i}, V_{2i} | S_{1i}, Y_i),
\end{aligned}
$$

where (a) follows since (X_i, Y_i, S_{1i}) is independent of $(X^{i-1}, S_1^{[k]\setminus\{i\}}, Y^{[k]\setminus\{i\}})$. For the last step, we define $V_{2i} := (M_{\mathsf{c},\{2\}}, X^{i-1}, S_{1,i+1}^k)$. Note that $(V_{\mathsf{c}i}, V_{2i}) \multimap X_i \multimap Y_i$ form a Markov chain. Second, we have

$$
\begin{aligned}
& k(R_{\mathsf{u},\{1,2\}} + R_{\mathsf{u},\{2\}}) \\
&\geq H(M_{\mathsf{u},\{1,2\}} | M_{\mathsf{c},\{1,2\}}, Y^k) + H(M_{\mathsf{u},\{2\}}) \\
&\overset{(a)}{\geq} H(S_1^k, M_{\mathsf{u},\{1,2\}} | M_{\mathsf{c},\{1,2\}}, Y^k) + H(M_{\mathsf{u},\{2\}}) - k\epsilon_k' \\
&= \sum_{i=1}^{k} H(S_{1i} | M_{\mathsf{c},\{1,2\}}, S_1^{i-1}, Y^k) + H(M_{\mathsf{u},\{1,2\}} | S_1^k, M_{\mathsf{c},\{1,2\}}, Y^k) + H(M_{\mathsf{u},\{2\}}) - k\epsilon_k' \\
&\geq \sum_{i=1}^{k} H(S_{1i} | V_{\mathsf{c}i}, Y_i) + H(M_{\mathsf{u},\{1,2\}}, M_{\mathsf{u},\{2\}} | S_1^k, M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}, Y^k) - k\epsilon_k' \\
&\overset{(b)}{\geq} \sum_{i=1}^{k} H(S_{1i} | V_{\mathsf{c}i}, Y_i) + H(S_2^k, M_{\mathsf{u},\{1,2\}}, M_{\mathsf{u},\{2\}} | S_1^k, M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}, Y^k) - k(\epsilon_k' + \epsilon_k'') \\
&\geq \sum_{i=1}^{k} H(S_{1i} | V_{\mathsf{c}i}, Y_i) + H(S_2^k | S_1^k, M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}, Y^k) - k(\epsilon_k' + \epsilon_k'') \\
&= \sum_{i=1}^{k} H(S_{1i} | V_{\mathsf{c}i}, Y_i) + \sum_{i=1}^{k} H(S_{2i} | S_2^{i-1}, S_1^k, M_{\mathsf{c},\{1,2\}}, M_{\mathsf{c},\{2\}}, Y^k) - k(\epsilon_k' + \epsilon_k'') \\
&\geq \sum_{i=1}^{k} H(S_{1i} | V_{\mathsf{c}i}, Y_i) + \sum_{i=1}^{k} H(S_{2i} | S_{1i}, V_{2i}, V_{\mathsf{c}i}, Y_i) - k(\epsilon_k' + \epsilon_k''),
\end{aligned}
$$

where (a) and (b) follow from the data processing inequality and Fano's inequality. The rest of the converse proof follows from the standard time-sharing argument, letting $k \to \infty$, and the fact that

$$
\begin{aligned}
& I(f_1(X,Y); V_{\mathsf{c}} | Y) + I(X; V_{\mathsf{c}}, V_2 | f_1(X,Y), Y) \\
&= I(X; V_{\mathsf{c}} | Y) + I(X; V_2 | f_1(X,Y), V_{\mathsf{c}}, Y).
\end{aligned}
$$

Finally, the cardinality bounds on $\mathcal{V}_{\mathsf{c}}$ and $\mathcal{V}_2$ can be proved using the convex cover method [5, Appendix C]. ■

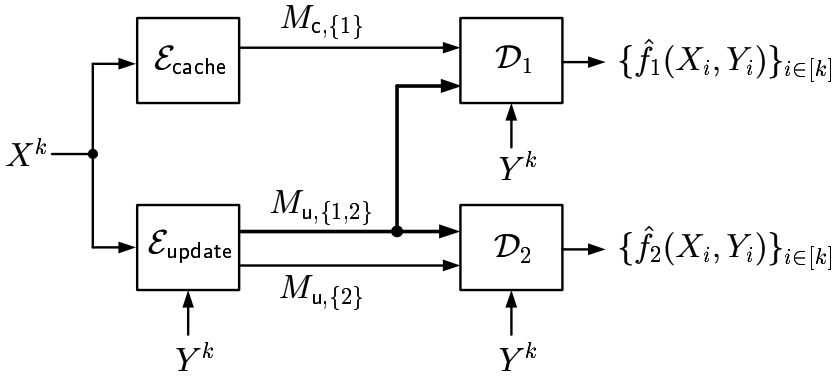

Figure 4.5: The source network with Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{1\}|\{2\},\{1,2\})$.

4.4 Configuration $(\{1\}|\{2\},\{1,2\})$

In this configuration (see Figure 4.5), User 1 has a private cache, User 2 has a private update, and additionally they both receive a common update. Again, there is a conflict at Decoder 1 because it receives both private cache and common update. Unfortunately, the optimal rate region $\mathcal{R}^*(\{1\}|\{2\},\{1,2\})$ is unknown.

Let us first consider the special case Configuration $(\{1\}|\{1,2\})$ to gain some insight. By symmetry, Configuration $(\{1\}|\{1,2\})$ is also a special case of the problem of sequential successive refinement addressed in Section 4.3. Thus, $\mathcal{R}^*(\{1\}|\{1,2\})$ is the set of rate tuples $\mathbf{R}$ such that $R_{\mathsf{c},\{2\}} = R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$,

$$\begin{aligned} R_{\mathsf{c},\{1\}} &\geq I(X; V|f_2(X,Y), Y), \\ R_{\mathsf{u},\{1,2\}} &\geq H(f_2(X,Y)|Y) + H(f_1(X,Y)|V, f_2(X,Y), Y), \end{aligned}$$

for some conditional pmf $p_{V|X}$. From the rate expressions, we can observe the following: First, Decoder 2 must recover the desired function f_2 from the single message $M_{\mathsf{u},\{1,2\}}$, which is also received by Decoder 1. Thus, it implies that Decoder 1 can also recover f_2 and use it as side information. Second, there are two different descriptions sent though the common link. From the perspective of Decoder 1, the description about f_2 is reconstructed *before* recovering the private cache content and the description about f_1 is reconstructed *after* recovering the private cache content. Therefore, in the achievability we can see both principles "common → private" and "cache → update". Unfortunately, currently there is no concrete example showing that both principles are required at the same time.

Based on the above observation, we present an inner bound and an outer bound that uses both principles "common → private" and "cache → update".

Proposition 4.1 (Inner Bound)**.** *A rate tuple* $\mathbf{R}$ *belongs to* $\mathcal{R}^*(\{1\}|\{2\},\{1,2\})$ *if its elements satisfy* $R_{\mathsf{c},\{2\}} = R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = 0$*, and*

$$\begin{aligned} R_{\mathsf{c},\{1\}} &> I(X; V|Y) - I(W; V|Y), \\ R_{\mathsf{u},\{1,2\}} &> I(V, X; W|Y) + H(f_1(X,Y)|W, V, Y), \\ R_{\mathsf{u},\{2\}} &> H(f_2(X,Y)|W, Y), \end{aligned}$$

for some conditional pmf $p_{V|X}p_{W|V,X,Y}$.

Note that by setting $W = f_2(X,Y)$, we recover the rate region $\mathcal{R}^*(\{1\}|\{1,2\})$.

Proof: Fix the conditional pmf $p_{V|X}p_{W|V,X,Y}$. Denote $s_{1i} = f_1(x_i,y_i)$ and $s_{2i} = f_2(x_i,y_i)$, $i \in [k]$. Assume that $\epsilon > \epsilon' > 0$.

Codebook generation: Randomly and independently generate $\lceil 2^{kR_{\mathsf{c},\{1\}}}\rceil\lceil 2^{kR_1}\rceil$ sequences $v^k(\ell_{\mathsf{v}},\ell_1)$, $\ell_{\mathsf{v}} \in [2^{kR_{\mathsf{c},\{1\}}}]$ and $\ell_1 \in [2^{kR_1}]$, each according to $\prod_{i=1}^k p_V(v_i)$. Also, for each $y^k \in \mathcal{Y}^k$, randomly and independently generate $\lceil 2^{kR_2}\rceil$ sequences $w^k(\ell_2,y^k)$, $\ell_2 \in [2^{kR_2}]$, each according to $\prod_{i=1}^k p_{W|Y}(w_i|y_i)$. Finally, for each $j \in \{1,2\}$, randomly and independently assign a bin index $m_j(s_j^k)$ to each sequence $s_j^k \in \mathcal{S}_j^k$ according to a uniform pmf over $[2^{kR_{2+j}}]$. The codebooks are revealed to all nodes.

Encoding: Upon seeing x^k, the cache encoder finds an index pair $(\ell_{\mathsf{v}},\ell_1)$ such that $(x^k, v^k(\ell_{\mathsf{v}},\ell_1)) \in T_{\epsilon'}^{(k)}(X,V)$. If there is more than one such index pair, it selects the one that minimizes $\ell_{\mathsf{v}}\lceil 2^{kR_1}\rceil + \ell_1$. If there is no such index pair, it sets $(\ell_{\mathsf{v}},\ell_1) = (1,1)$. Then, the cache encoder sends the index ℓ_{v} to Decoder 1.

Upon seeing (x^k,y^k), the update encoder first finds the same sequence $v^k(\ell_{\mathsf{v}},\ell_1)$ as the cache encoder. Then, the update encoder finds an index ℓ_2 such that $(v^k(\ell_{\mathsf{v}},\ell_1), x^k, w^k(\ell_2,y^k)) \in T_{\epsilon}^{(k)}(V,X,W|y^k)$. If there is more than one such index, it selects the smallest one. If there is no such index, it sets $\ell_2 = 1$. Then, the update encoder sends the indices $(\ell_2, m_1(s_1^k))$ to Decoders 1 and 2 through the common update link. Finally, the update encoder sends the index $m_2(s_2^k)$ to Decoder 2 through the private update link. Therefore, we have the conditions $R_{\mathsf{u},\{1,2\}} \geq R_2 + R_3$ and $R_{\mathsf{u},\{2\}} \geq R_4$.

Decoding: Upon seeing $(\ell_{\mathsf{v}},\ell_2,m_1)$, Decoder 1 finds the unique index $\hat{\ell}_1$ such that $(w^k(\ell_2,y^k), v^k(\ell_{\mathsf{v}},\hat{\ell}_1), y^k) \in \mathcal{T}_{\epsilon}^{(k)}(W,V,Y)$; otherwise it sets $\hat{\ell}_1 = 1$. Then, Decoder 1 declares the estimate $\hat{s}_1^k$ if it is the unique sequence with bin index m_1 such that $(\hat{s}_1^k, w^k(\ell_2,y^k), v^k(\ell_{\mathsf{v}},\hat{\ell}_1), y^k) \in \mathcal{T}_{\epsilon}^{(k)}(S_1,W,V,Y)$; otherwise it declares an error.

Upon seeing (ℓ_2,m_1,m_2), Decoder 2 declares the estimate $\hat{s}_2^k$ if it is the unique sequence with bin index m_2 such that $(\hat{s}_2^k, w^k(\ell_2,y^k), y^k) \in \mathcal{T}_{\epsilon}^{(k)}(S_2,W,Y)$; otherwise it declares an error.

Analysis: The following can be shown using the standard typicality arguments. Note that each event is conditioned on the success of the previous events.

1. If $R_{\mathsf{c},\{1\}} + R_1 > I(X;V) + \delta(\epsilon')$, then the cache encoder finds an index pair $(\ell_{\mathsf{v}},\ell_1)$ with high probability (w.h.p.). Note that it implies that $(x^k,y^k,v^k(\ell_{\mathsf{v}},\ell_1)) \in T_{\epsilon}^{(k)}(X,Y,V)$ w.h.p..

2. If $R_1 < I(W,Y;V) + \delta(\epsilon)$, then Decoder 1 identifies the index ℓ_1 w.h.p..

3. If $R_2 > I(V,X;W|Y) + \delta(\epsilon)$, then the update encoder finds an index ℓ_2 w.h.p..

4. If $R_3 > H(f_1(X,Y)|W,V,Y) + \delta(\epsilon)$, then Decoder 1 recovers s_1^k correctly w.h.p..

5. If $R_4 > H(f_2(X,Y)|W,Y) + \delta(\epsilon)$, then Decoder 2 recovers s_2^k correctly w.h.p..

The rest of the proof follows from the Fourier–Motzkin elimination and then by letting $\epsilon, \epsilon' \to 0$. ■

Proposition 4.2 (Outer Bound)**.** *If* $\mathbf{R} \in \mathcal{R}^*(\{1\}|\{2\},\{1,2\})$*, then its elements must satisfy* $R_{\mathsf{c},\{2\}} = R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = 0$ *and the inequalities*

$$\begin{aligned} R_{\mathsf{c},\{1\}} &> I(X;V|W,Y), \\ R_{\mathsf{u},\{1,2\}} &> I(X;W|Y) + H(f_1(X,Y)|W,V,Y), \\ R_{\mathsf{u},\{2\}} &> H(f_2(X,Y)|W,Y), \end{aligned}$$

for some conditional pmf $p_{V|X}p_{W|V,X,Y}$*.*

The proof of Proposition 4.2 follows similar lines as the converse proof of Theorem 3.1 and is thus omitted. We remark that the outer bound in Proposition 4.2 can be expressed as

$$\begin{aligned} R_{\mathsf{c},\{1\}} &> I(X;V|Y) - I(W;V|Y) + I(W;V|X,Y), \\ R_{\mathsf{u},\{1,2\}} &> I(V,X;W|Y) + H(f_1(X,Y)|W,V,Y) - I(W;V|X,Y), \\ R_{\mathsf{u},\{2\}} &> H(f_2(X,Y)|W,Y). \end{aligned}$$

Thus, the inner bound and the outer bound coincide if all the extreme points can be achieved by some conditional pmf $p_{V|X}p_{W|V,X,Y}$ satisfying $I(W;V|X,Y) = 0$.

4.5 Configuration $(\{1\},\{2\}|\{1,2\})$

Configuration $(\{1\},\{2\}|\{1,2\})$ (see Figure 4.6) is essentially the setup studied by Maddah-Ali and Niesen [26], where each user has a private cache and both receive a common update. Similar to Section 4.4, we provide an achievable scheme that uses both principles "common → private" and "cache → update".

Proposition 4.3 (Inner Bound)**.** *A rate tuple* $\mathbf{R}$ *belongs to* $\mathcal{R}^*(\{1\},\{2\}|\{1,2\})$ *if its elements satisfy* $R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$*, and*

$$\begin{aligned} R_{\mathsf{u},\{1,2\}} &> I(V_1,V_2,X;W|Y) + \max_{j\in\{1,2\}} I(V_{3-j},X;U|V_j,W,Y) \\ &\quad + H(f_1(X,Y)|U,V_1,W,Y) + H(f_2(X,Y)|U,V_2,W,Y), \\ R_{\mathsf{c},\{1\}} &> I(X;V_1|Y) - I(W;V_1|Y), \\ R_{\mathsf{c},\{2\}} &> I(X;V_2|Y) - I(W;V_2|Y), \\ R_{\mathsf{c},\{1\}} + R_{\mathsf{c},\{2\}} &> I(X;V_1|Y) + I(X;V_2|Y) + I(V_1;V_2|X) \\ &\quad - I(W;V_1|Y) - I(W;V_2|Y), \end{aligned}$$

for some conditional pmf $p_{V_1,V_2|X}p_{U,W|V_1,V_2,X,Y}$*.*

Proof: Fix the conditional pmf $p_{V_1,V_2|X}p_{U,W|V_1,V_2,X,Y}$. Denote $s_{1i} = f_1(x_i,y_i)$ and $s_{2i} = f_2(x_i,y_i)$, $i \in [k]$. Assume that $\epsilon > \epsilon' > 0$.

Codebook generation: For each $j \in \{1,2\}$, randomly and independently generate $\lceil 2^{kR_{\mathsf{c},\{j\}}} \rceil \lceil 2^{kR_j} \rceil$ sequences $v_j^k(\ell_j,\tilde{\ell}_j)$, $\ell_j \in [2^{kR_{\mathsf{c},\{j\}}}]$ and $\tilde{\ell}_j \in [2^{kR_j}]$, each according to $\prod_{i=1}^k p_{V_j}(v_{ji})$. Next, For each $y^k \in \mathcal{Y}^k$, randomly and independently generate $\lceil 2^{kR_{\mathsf{w}}} \rceil$ sequences $w^k(\ell_{\mathsf{w}},y^k)$, $\ell_{\mathsf{w}} \in [2^{kR_{\mathsf{w}}}]$, each according to $\prod_{i=1}^k p_{W|Y}(w_i|y_i)$. Then, for each pair $(\ell_{\mathsf{w}},y^k) \in [2^{kR_{\mathsf{w}}}] \times \mathcal{Y}^k$, randomly and independently generate $\lceil 2^{kR_{\mathsf{u}}} \rceil \lceil 2^{kR_0} \rceil$ sequences $u^k(\ell_{\mathsf{u}},\tilde{\ell}_0,\ell_{\mathsf{w}},y^k)$, where $\ell_{\mathsf{u}} \in [2^{kR_{\mathsf{u}}}]$ and $\tilde{\ell}_0 \in [2^{kR_0}]$, each

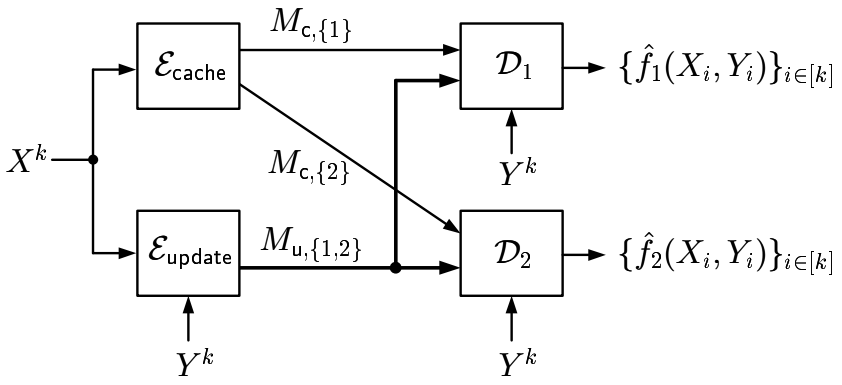

Figure 4.6: The source network with Configuration $(\mathcal{A}_\mathsf{c}|\mathcal{A}_\mathsf{u}) = (\{1\},\{2\}|\{1,2\})$.

according to $\prod_{i=1}^k p_{U|W,Y}(u_i|w_i,y_i)$. Finally, for each $j\in\{1,2\}$, randomly and independently assign a bin index $m_j(s_j^k)$ to each sequence $s_j^k\in\mathcal{S}_j^k$ according to a uniform pmf over $[2^{kR_{2+j}}]$. The codebooks are revealed to all nodes.

Encoding: Upon seeing x^k, the cache encoder finds an index tuple $(\ell_1,\tilde{\ell}_1,\ell_2,\tilde{\ell}_2)$ such that $(x^k,v_1^k(\ell_1,\tilde{\ell}_1),v_2^k(\ell_2,\tilde{\ell}_2))\in T_{\epsilon'}^{(k)}(X,V_1,V_2)$. If there is more than one such index tuple, it selects one following an arbitrary rule coordinated with the update encoder. If there is no such index tuple, it sets $(\ell_1,\tilde{\ell}_1,\ell_2,\tilde{\ell}_2)=(1,1,1,1)$. Finally, the cache encoder sends the index ℓ_j to Decoder j, where $j\in\{1,2\}$.

Upon seeing (x^k,y^k), the update encoder first finds the same sequences $v_1^k(\ell_1,\tilde{\ell}_1)$ and $v_2^k(\ell_2,\tilde{\ell}_2)$ as done by the cache encoder. Then, the update encoder finds an index ℓ_w such that $(x^k,v_1^k(\ell_1,\tilde{\ell}_1),v_2^k(\ell_2,\tilde{\ell}_2),w^k(\ell_\mathsf{w},y^k))\in T_\epsilon^{(k)}(X,V_1,V_2,W|y^k)$. If there is more than one such index, it selects the smallest one. If there is no such index, it sets $\ell_\mathsf{w}=1$. Next, the update encoder finds an index pair $(\ell_\mathsf{u},\tilde{\ell}_0)$ such that $(x^k,v_1^k(\ell_1,\tilde{\ell}_1),v_2^k(\ell_2,\tilde{\ell}_2),u^k(\ell_\mathsf{u},\tilde{\ell}_0,\ell_\mathsf{w},y^k))\in T_\epsilon^{(k)}(X,V_1,V_2,U|w^k,y^k)$. If there is more than one such index pair, it selects the one that minimizes $\ell_\mathsf{u}\lceil 2^{kR_0}\rceil+\tilde{\ell}_0$. If there is no such index pair, it sets $(\ell_\mathsf{u},\tilde{\ell}_0)=(1,1)$.

Finally, the update encoder sends the index tuple $(\ell_\mathsf{w},\ell_\mathsf{u},m_1(s_1^k),m_2(s_2^k))$ to Decoders 1 and 2 through the common link. Therefore, we have the condition $R_{\mathsf{u},\{1,2\}}\geq R_\mathsf{w}+R_\mathsf{u}+R_3+R_4$.

Decoding: Consider $j\in\{1,2\}$. Upon seeing $(\ell_j,\ell_\mathsf{w},\ell_\mathsf{u},m_1,m_2)$, Decoder j first recovers $w^k(\ell_\mathsf{w},y^k)$. Then, Decoder j finds the unique index $\hat{\ell}_j$ such that $(v_j^k(\ell_j,\hat{\ell}_j),w^k(\ell_\mathsf{w},y^k),y^k)\in\mathcal{T}_\epsilon^{(k)}(V_j,W,Y)$; otherwise it sets $\hat{\ell}_j=1$. Next, Decoder j finds the unique index $\hat{\ell}_\mathsf{u}$ such that $(u^k(\ell_\mathsf{u},\hat{\ell}_0,\ell_\mathsf{w},y^k),v_j^k(\ell_j,\hat{\ell}_j),w^k(\ell_\mathsf{w},y^k),y^k)\in\mathcal{T}_\epsilon^{(k)}(U,V_j,W,Y)$. Finally, Decoder j declares the estimate $\hat{s}_j^k$ if it is the unique sequence with bin index m_j such that $(\hat{s}_j^k,u^k(\ell_\mathsf{u},\hat{\ell}_0,\ell_\mathsf{w},y^k),v_j^k(\ell_j,\hat{\ell}_j),w^k(\ell_\mathsf{w},y^k),y^k)\in\mathcal{T}_\epsilon^{(k)}(S_j,U,V_j,W,Y)$; otherwise it declares an error.

Analysis: The following can be shown using the standard typicality arguments. Note that each event is conditioned on the success of the previous events.

1. If it holds that

$$R_{\mathsf{c},\{1\}}+R_1>I(X;V_1)+\delta(\epsilon'),$$

$$R_{\mathsf{c},\{2\}} + R_2 > I(X;V_2) + \delta(\epsilon'),$$
$$R_{\mathsf{c},\{1\}} + R_1 + R_{\mathsf{c},\{2\}} + R_2 > I(X;V_1) + I(X;V_2) + I(V_1;V_2|X) + \delta(\epsilon'),$$

then the cache encoder finds an index tuple $(\ell_1, \tilde{\ell}_1, \ell_2, \tilde{\ell}_2)$ w.h.p.. Note that it implies that $(x^k, y^k, v_1^k(\ell_1, \tilde{\ell}_1), v_2^k(\ell_2, \tilde{\ell}_2)) \in T_\epsilon^{(k)}(X, Y, V_1, V_2)$ w.h.p..

2. If $R_{\mathsf{w}} > I(V_1, V_2, X; W|Y) + \delta(\epsilon)$, then the update encoder finds an index ℓ_{w} w.h.p..

3. For $j \in \{1, 2\}$, if $R_j < I(W, Y; V_j) + \delta(\epsilon)$, then Decoder j identifies the index $\tilde{\ell}_j$ w.h.p..

4. If $R_{\mathsf{u}} + R_0 > I(V_1, V_2, X; U|W, Y) + \delta(\epsilon)$, then the update encoder finds an index pair $(\ell_{\mathsf{u}}, \tilde{\ell}_0)$ w.h.p..

5. For $j \in \{1, 2\}$, if $R_0 < I(V_j; U|W, Y) + \delta(\epsilon)$, then Decoder j identifies the index $\tilde{\ell}_0$ w.h.p..

6. For $j \in \{1, 2\}$, if $R_{2+j} > H(f_j(X, Y)|U, V_j, W, Y) + \delta(\epsilon)$, then Decoder j recovers s_j^k correctly w.h.p..

The rest of the proof follows from the Fourier–Motzkin elimination and then by letting $\epsilon, \epsilon' \to 0$. ■

We remark that after Decoders 1 and 2 recover (w^k, v_1^k) and (w^k, v_2^k), respectively, the system can be treated as a special case of the Kaspi/Heegard–Berger problem with an informed encoder [20] (see also [21]). The rate region presented in Proposition 4.3 is not convex in general, but we can easily convexify it by introducing a time-sharing random variable Q.

Remark 4.1. *Let us briefly discuss the scenario in which the requests are only locally known, i.e., the users have no information about each other's requests. In that scenario, every configuration with at least one common link can be simplified to the Kaspi/Heegard–Berger problem by setting some of the rates to zero. Following the principle "cache → update," for each configuration we can develop an achievability with rate expressions similar to Proposition 4.3 with $W = \emptyset$. For example, consider Configuration $(\{1\}, \{2\}|\{1, 2\})$ in which the requests are only locally known. It can be shown that the following rate region is achievable: the set of rate tuples $\mathbf{R}$ whose elements satisfy $R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$, and*

$$\begin{aligned}
R_{\mathsf{c},\{1\}} &> I(X;V_1|Y_1),\\
R_{\mathsf{c},\{2\}} &> I(X;V_2|Y_2),\\
R_{\mathsf{c},\{1\}} + R_{\mathsf{c},\{2\}} &> I(X;V_1|Y_1) + I(X;V_2|Y_2) + I(V_1;V_2|X),\\
R_{\mathsf{u},\{1,2\}} &> \max\{I(V_2, Y_2, X; U|V_1, Y_1), I(V_1, Y_1, X; U|V_2, Y_2)\}\\
&\quad + H(f_1(X, Y)|U, V_1, Y_1) + H(f_2(X, Y)|U, V_2, Y_2),
\end{aligned}$$

for some conditional pmf $p_{V_1,V_2|X} p_{U|V_1,V_2,X,Y_1,Y_2}$.

Now we present an outer bound which has a similar form as the achievable rate region in Proposition 4.3 with $W = \emptyset$.

Proposition 4.4 (Outer Bound)**.** *If* $\mathbf{R} \in \mathcal{R}^*(\{1\},\{2\}|\{1,2\})$*, then its elements must satisfy* $R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$ *and the inequalities*

$$\begin{aligned} R_{\mathsf{c},\{1\}} &\geq I(X;V_1|Y), && (4.2)\\ R_{\mathsf{c},\{2\}} &\geq I(X;V_2|Y), && (4.3)\\ R_{\mathsf{u},\{1,2\}} &\geq \max\{I(X;U|V_1,Y), I(X;U|V_2,Y)\} \\ &\quad + H(f_1(X,Y)|U,V_1,Y) + H(f_2(X,Y)|U,V_2,Y), \end{aligned}$$

for some conditional pmf $p_{V_1,V_2|X}p_{U|V_1,V_2,X,Y}$ *such that*

$$I(X;V_1|Y) + I(X;V_2|Y) \geq I(X;V_1,V_2|Y). \tag{4.4}$$

Proof: Denote $S_{1i} = f_1(X_i,Y_i)$ and $S_{2i} = f_2(X_i,Y_i)$ for $i \in [k]$. First, for each $j \in \{1,2\}$, we have

$$\begin{aligned} kR_{\mathsf{c},\{j\}} &\geq H(M_{\mathsf{c},\{j\}}|Y^k) \\ &= I(X^k; M_{\mathsf{c},\{j\}}|Y^k) \\ &= \sum_{i=1}^k I(X_i; M_{\mathsf{c},\{j\}}|X^{i-1},Y^k) \\ &= \sum_{i=1}^k I(X_i; M_{\mathsf{c},\{j\}},X^{i-1},Y^{[k]\setminus\{i\}}|Y_i) \\ &= \sum_{i=1}^k I(X_i;V_{ji}|Y_i). \end{aligned}$$

The last step follows by defining $V_{ji} = (M_{\mathsf{c},\{j\}}, X^{i-1}, Y^{[k]\setminus\{i\}})$, $j \in \{1,2\}$, for all $i \in [k]$. Note that for $j \in \{1,2\}$, $V_{ji} \multimap X_i \multimap Y_i$ form a Markov chain. Next, for $j \in \{1,2\}$, we have

$$\begin{aligned} kR_{\mathsf{u},\{1,2\}} &\geq H(M_{\mathsf{u},\{1,2\}}|M_{\mathsf{c},\{j\}},Y^k) \\ &= I(X^k; M_{\mathsf{u},\{1,2\}}|M_{\mathsf{c},\{j\}},Y^k) \\ &= \sum_{i=1}^k I(X_i; M_{\mathsf{u},\{1,2\}}|X^{i-1},M_{\mathsf{c},\{j\}},Y^k) \\ &= \sum_{i=1}^k I(X_i; M_{\mathsf{u},\{1,2\}},X^{i-1},Y^{[k]\setminus\{i\}}|V_{ji},Y_i) \\ &= \sum_{i=1}^k I(X_i;U_i|V_{ji},Y_i). \end{aligned} \tag{4.5}$$

The last step follows by defining $U_i = (M_{\mathsf{u},\{1,2\}}, X^{i-1}, Y^{[k]\setminus\{i\}})$, for all $i \in [k]$. From the data processing inequality and Fano's inequality, we have for $j \in \{1,2\}$,

$$\begin{aligned} k\epsilon_{jk} &\geq H(S_j^k|M_{\mathsf{c},\{j\}},M_{\mathsf{u},\{1,2\}},Y^k) \\ &= \sum_{i=1}^k H(S_{ji}|S_j^{i-1},M_{\mathsf{c},\{1\}},M_{\mathsf{u},\{1,2\}},Y^k) \end{aligned}$$

$$\geq \sum_{i=1}^{k} H(S_{ji}|X^{i-1}, M_{\mathsf{c},\{1\}}, M_{\mathsf{u},\{1,2\}}, Y^k)$$
$$= \sum_{i=1}^{k} H(S_{ji}|U_i, V_{ji}, Y_i). \tag{4.6}$$

Thus, the inequalities (4.5) and (4.6) imply that for $j \in \{1,2\}$,

$$k(R_{\mathsf{u},\{1,2\}} + \epsilon_{1k} + \epsilon_{2k})$$
$$\geq \sum_{i=1}^{k} \left[I(X_i; U_i|V_{ji}, Y_i) + H(S_{1i}|U_i, V_{1i}, Y_i) + H(S_{2i}|U_i, V_{2i}, Y_i) \right].$$

Finally, following a proof step in [42, Theorem 3], we have

$$\sum_{i=1}^{k} I(X_i; V_{1i}|Y_i) + \sum_{i=1}^{k} I(X_i; V_{2i}|Y_i)$$
$$= \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{1\}}, X^{i-1}, Y^{[k]\setminus\{i\}}|Y_i) + \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{2\}}, X^{i-1}, Y^{[k]\setminus\{i\}}|Y_i)$$
$$= \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{1\}}|X^{i-1}, Y^k) + \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{2\}}|X^{i-1}, Y^k)$$
$$= I(X^k; M_{\mathsf{c},\{1\}}|Y^k) + I(X^k; M_{\mathsf{c},\{2\}}|Y^k)$$
$$= H(M_{\mathsf{c},\{1\}}|Y^k) + H(M_{\mathsf{c},\{2\}}|Y^k)$$
$$\geq H(M_{\mathsf{c},\{1\}}, M_{\mathsf{c},\{2\}}|Y^k)$$
$$= I(X^k; M_{\mathsf{c},\{1\}}, M_{\mathsf{c},\{2\}}|Y^k)$$
$$= \sum_{i=1}^{k} I(X_i; M_{\mathsf{c},\{1\}}, M_{\mathsf{c},\{2\}}|X^{i-1}, Y^k)$$
$$= \sum_{i=1}^{k} I(X_i; V_{1i}, V_{2i}|Y_i).$$

The rest of the proof follows from the standard time-sharing argument and then letting $k \to \infty$. ■

The outer bound in Proposition 4.4 can be relaxed to the following cut-set based bound.

Corollary 4.1. *If* $\mathbf{R} \in \mathcal{R}^*(\{1\},\{2\}|\{1,2\})$*, then its elements must satisfy* $R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$ *and the inequalities*

$$\begin{aligned} R_{\mathsf{c},\{1\}} &\geq I(X; V_1|Y), \\ R_{\mathsf{c},\{2\}} &\geq I(X; V_2|Y), \\ R_{\mathsf{c},\{1\}} + R_{\mathsf{c},\{2\}} &\geq I(X; V_1, V_2|Y), \\ R_{\mathsf{u},\{1,2\}} &\geq \max\{H(f_1(X,Y)|V_1, Y), H(f_2(X,Y)|V_2, Y), \\ &\qquad H(f_1(X,Y), f_2(X,Y)|V_1, V_2, Y)\}, \end{aligned}$$

for some conditional pmf $p_{V_1,V_2|X}$.

Proof: First, note that (4.2), (4.3), and (4.4) imply that $R_{\mathsf{c},\{1\}} + R_{\mathsf{c},\{2\}} \geq I(X;V_1,V_2|Y)$. Next, we show that given any conditional pmf $p_{V_1,V_2|X}p_{U|V_1,V_2,X,Y}$ that satisfies (4.4), it holds that

$$\begin{aligned}&\max_{j\in\{1,2\}}\{I(X;U|V_j,Y)\} + H(f_1(X,Y)|U,V_1,Y) + H(f_2(X,Y)|U,V_2,Y)\\ &\geq \max\{H(f_1(X,Y)|V_1,Y), H(f_2(X,Y)|V_2,Y), H(f_1(X,Y),f_2(X,Y)|V_1,V_2,Y)\}.\end{aligned}$$

Indeed, we have for $\ell \in \{1,2\}$,

$$\begin{aligned}&\max_{j\in\{1,2\}}\{I(X;U|V_j,Y)\} + H(f_1(X,Y)|U,V_1,Y) + H(f_2(X,Y)|U,V_2,Y)\\ &\geq I(X;U|V_\ell,Y) + H(f_\ell(X,Y)|U,V_\ell,Y)\\ &\geq I(f_\ell(X,Y);U|V_\ell,Y) + H(f_\ell(X,Y)|U,V_\ell,Y)\\ &= H(f_\ell(X,Y)|V_\ell,Y).\end{aligned}$$

Also, we have

$$\begin{aligned}&\max_{j\in\{1,2\}}\{I(X;U|V_j,Y)\} + H(f_1(X,Y)|U,V_1,Y) + H(f_2(X,Y)|U,V_2,Y)\\ &\geq \max_{j\in\{1,2\}}\{I(X;U|V_j,Y)\} + H(f_1(X,Y),f_2(X,Y)|U,V_1,V_2,Y)\\ &\overset{(a)}{\geq} I(X;U|V_1,V_2,Y) + H(f_1(X,Y),f_2(X,Y)|U,V_1,V_2,Y)\\ &\geq I(f_1(X,Y),f_2(X,Y);U|V_1,V_2,Y) + H(f_1(X,Y),f_2(X,Y)|U,V_1,V_2,Y)\\ &= H(f_1(X,Y),f_2(X,Y)|V_1,V_2,Y),\end{aligned}$$

where (a) follows from (4.4). Finally, we remove the constraints (4.4) and the desired corollary is established. ■

Now let us look at two examples of Configuration $(\{1\},\{2\}|\{1,2\})$.

Example 4.1 (Independent Selection)**.** *Let $X = (A,B)$, $Y = (Y_1,Y_2)$, where A,B,Y_1,Y_2 are i.i.d. drawn from Bernoulli$(1/2)$. Assume that*

$$f_j(X,Y) = \begin{cases} A & \text{if } Y_j = 0,\\ B & \text{if } Y_j = 1,\end{cases}$$

where $j \in \{1,2\}$. Unfortunately, the optimal rate region is unknown, even for the symmetric case, i.e., $R_{\mathsf{c},\{1\}} = R_{\mathsf{c},\{2\}} = R_{\mathsf{c}}$. Denote $R_{\mathsf{u}} = R_{\mathsf{u},\{1,2\}}$. It can be easily checked that $(R_{\mathsf{c}},R_{\mathsf{u}}) = (0,1.5),(2,0)$ are two extreme points of the symmetry-constrained optimal rate region. Next, borrowing the idea from the achievability in [26, Example 4], we substitute $W = \emptyset$, $V_1 = A$, $V_2 = B$, and

$$U = \begin{cases} A & \text{if } (Y_1,Y_2) = (0,0),\\ \emptyset & \text{if } (Y_1,Y_2) = (0,1),\\ A\oplus B & \text{if } (Y_1,Y_2) = (1,0),\\ B & \text{if } (Y_1,Y_2) = (1,1),\end{cases}$$

into the rate expressions of Proposition 4.3. Then, the rate pair $(R_{\mathsf{c}},R_{\mathsf{u}}) = (1,0.5)$ is achievable. The inner bound plotted in solid blue in Figure 4.7 follows by time

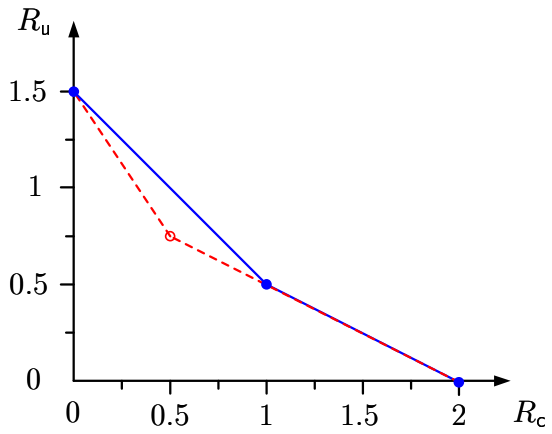

Figure 4.7: The inner and outer bounds for the example of independent selection (Example 4.1). The inner bound is plotted in solid blue. The outer bound is plotted in dashed red.

sharing among $(0, 1.5)$, $(1, 0.5)$, *and* $(2, 0)$. *On the other hand, the outer bound in Corollary 4.1 can be relaxed by only considering the intersection of*

$$\{(R_c, R_u) : R_c \geq I(X; V_1|Y), R_u \geq H(f_1(X, Y)|V_1, Y) \text{ for some } p_{V_1|X}\}$$

and

$$\{(R_c, R_u) : 2R_c \geq I(X; V_1, V_2|Y), \\ R_u \geq H(f_1(X, Y), f_2(X, Y)|V_1, V_2, Y) \text{ for some } p_{V_1,V_2|X}\}.$$

Then, following similar steps as the proof of Proposition 3.1, we have that for any $R_c \geq 0$,

$$R_u \geq \max\left\{\left(1 - \frac{R_c}{2}\right)^+, \left(\frac{3}{2} - \frac{3}{2}R_c\right)^+\right\}.$$

The above outer bound is plotted in dashed red in Figure 4.7.

Example 4.2 (Complementary Selection)**.** *Let* A, B, Y *be i.i.d. Bernoulli*$(1/2)$ *random variables and* $X = (A, B)$. *Assume that*

$$(f_1(X, Y), f_2(X, Y)) = \begin{cases} (A, B) & \text{if } Y = 0, \\ (B, A) & \text{if } Y = 1. \end{cases}$$

That is, the desired components of the two users are always complementary to each other. Again, the optimal rate region is unknown, even for the symmetric case, i.e., $R_{c,\{1\}} = R_{c,\{2\}} = R_c$. *Denote* $R_u = R_{u,\{1,2\}}$. *It is easy to see that* $(R_c, R_u) = (0, 2), (2, 0)$ *are two extreme points of the symmetry-constrained optimal rate region. Next, we consider the following choice of auxiliary random variables, which borrows the idea from the achievability in [26, Appendix]. Let* A_2, B_1 *be i.i.d. Bernoulli(q)*

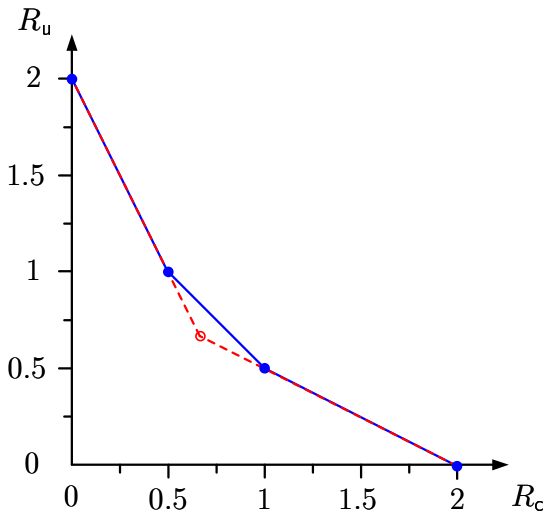

Figure 4.8: The inner and outer bounds for the example of complementary selection (Example 4.2). The inner bound is plotted in solid blue. The outer bound is plotted in dashed red.

and denote $A_1 = A \oplus A_2$, $B_2 = B \oplus B_1$. *We substitute* $W = \emptyset$, $V_1 = A_1 \oplus B_1$, $V_2 = A_2 \oplus B_2$, *and*

$$U = \begin{cases} (A_2, B_1) & \text{if } Y = 0, \\ (A_1, B_2) & \text{if } Y = 1, \end{cases}$$

into the rate expressions of Proposition 4.3. It can be verified that the rate pair $(R_{\mathsf{c}}, R_{\mathsf{u}}) = (1, 0.5)$ *is achievable with* $q = 0$ *and* $(R_{\mathsf{c}}, R_{\mathsf{u}}) = (0.5, 1)$ *is achievable with* $q = 1/2$. *We remark that when* $q = 1/2$, *we have* $I(V_1, V_2|X) = 1$, *i.e.,* V_1 *and* V_2 *are not conditionally independent given* X. *The inner bound plotted in solid blue in Figure 4.8 follows by time sharing among* $(0, 2)$, $(0.5, 1)$, $(1, 0.5)$, *and* $(2, 0)$. *Similar to Example 4.1, the outer bound in Corollary 4.1 implies that for any* $R_{\mathsf{c}} \geq 0$,

$$R_{\mathsf{u}} \geq \max\left\{\left(1 - \frac{R_{\mathsf{c}}}{2}\right)^+, (2 - 2R_{\mathsf{c}})^+\right\}.$$

The above outer bound is plotted in dashed red in Figure 4.8.

To end this chapter, we present a general achievable scheme involving all six rate components, which includes all achievable schemes in the previous sections as special cases. The achievable scheme follows by a straightforward extension of Proposition 4.3.

Proposition 4.5 (Inner Bound)**.** *A rate tuple* $\mathbf{R}$ *belongs to* $\mathcal{R}^*$ *if its elements satisfy*

$$R_{\mathsf{c},\{1,2\}} > I(X; V_{\mathsf{c}}|Y), \tag{4.7}$$

$$R_{\mathsf{u},\{1,2\}} > I(V_1, V_2, X; W|V_{\mathsf{c}}, Y) + \max_{j\in\{1,2\}} I(V_{3-j}, X; U|V_j, W, V_{\mathsf{c}}, Y), \tag{4.8}$$

$$R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{1\}} > I_{\mathsf{c}} + I(X; V_1|V_{\mathsf{c}}, Y) - I(W; V_1|V_{\mathsf{c}}, Y), \tag{4.9}$$

$$R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{2\}} > I_{\mathsf{c}} + I(X; V_2|V_{\mathsf{c}}, Y) - I(W; V_2|V_{\mathsf{c}}, Y), \tag{4.10}$$

$$\begin{aligned} R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{1\}} + R_{\mathsf{c},\{2\}} > {} & I_{\mathsf{c}} + I(X; V_1|V_{\mathsf{c}}, Y) + I(X; V_2|V_{\mathsf{c}}, Y) + I(V_1; V_2|X, V_{\mathsf{c}}) \\ & - I(W; V_1|V_{\mathsf{c}}, Y) - I(W; V_2|V_{\mathsf{c}}, Y), \end{aligned} \tag{4.11}$$

$$R_{\mathsf{u},\{1,2\}} + R_{\mathsf{u},\{1\}} > I_{\mathsf{u}} + H(f_1(X, Y)|U, V_1, W, V_{\mathsf{c}}, Y), \tag{4.12}$$

$$R_{\mathsf{u},\{1,2\}} + R_{\mathsf{u},\{2\}} > I_{\mathsf{u}} + H(f_2(X, Y)|U, V_2, W, V_{\mathsf{c}}, Y), \tag{4.13}$$

$$R_{\mathsf{u},\{1,2\}} + R_{\mathsf{u},\{1\}} + R_{\mathsf{u},\{2\}} > I_{\mathsf{u}} + \sum_{j=1}^{2} H(f_j(X, Y)|U, V_j, W, V_{\mathsf{c}}, Y), \tag{4.14}$$

for some conditional pmf $p_{V_{\mathsf{c}},V_1,V_2|X} p_{U,W|V_{\mathsf{c}},V_1,V_2,X,Y}$*, where*

$$\begin{aligned} I_{\mathsf{c}} &= I(X; V_{\mathsf{c}}|Y), \\ I_{\mathsf{u}} &= I(V_1, V_2, X; W|V_{\mathsf{c}}, Y) + \max_{j\in\{1,2\}} I(V_{3-j}, X; U|V_j, W, V_{\mathsf{c}}, Y). \end{aligned}$$

Proof Outline: Fix a conditional pmf $p_{V_{\mathsf{c}},V_1,V_2|X} p_{U,W|V_{\mathsf{c}},V_1,V_2,X,Y}$. The codebook generation is similar to Proposition 4.3, except the following two differences:

1. We generate an extra codebook to map each source sequence $x^k \in \mathcal{X}^k$ to a description v_{c}^k, each of which is assigned a bin index $m_{\mathsf{c},\{1,2\}}$;

2. The codebooks for the descriptions (u^k, v_1^k, v_2^k, w^k) in Proposition 4.3 are then generated by superimposing on the descriptions $\{v_{\mathsf{c}}^k\}$.

Then, the entire communication proceeds as follows. First, we apply Wyner–Ziv coding to convey the description v_{c}^k through the common cache link. After recovering the sequence v_{c}^k at the decoders, all nodes have v_{c}^k as side information. Then, we use the extended achievable scheme in Proposition 4.3 to convey (u^k, v_j^k, w^k) to Decoder j, $j \in \{1, 2\}$. Note that each private cache description v_j^k, $j \in \{1, 2\}$ can be split into two parts: one is sent through the common cache link and the other is sent through the private cache link. Similarly, each bin index for the sequence of functions $\{f_j(X_i, Y_i)\}_{i\in[k]}$ can be split into two parts: one is sent through the common update link and the other is sent through the private update link. ■

Finally, we summarize how to specialize Proposition 4.5 to recover the achievable schemes for the considered five configurations.

1. Theorem 4.1
   Set $R_{\mathsf{u},\{1,2\}} = 0$, $U = W = \emptyset$, and $p_{V_{\mathsf{c}},V_1,V_2|X} = p_{V_{\mathsf{c}}|X} p_{V_1|V_{\mathsf{c}},X} p_{V_2|V_{\mathsf{c}},X}$. Then, the inequalities (4.8), (4.11), and (4.14) become redundant. Finally, note that for $j \in \{1, 2\}$, the bound on sum rate $R_{\mathsf{c},\{1,2\}} + R_{\mathsf{c},\{j\}}$ can be relaxed to the bound on individual rate $R_{\mathsf{c},\{j\}}$.

2. Theorem 4.2
   Set $R_{\mathsf{c},\{1\}} = R_{\mathsf{c},\{2\}} = 0$ and $V_1 = V_2 = W = \emptyset$. Then, the inequalities (4.9), (4.10), (4.11) become redundant. Finally, note that the bounds on sum rate (4.12), (4.13), (4.14) can be relaxed to the bounds on the individual private update rate $R_{\mathsf{u},\{j\}}$, $j \in \{1, 2\}$.

3. Theorem 4.3
   Set $R_{\mathsf{c},\{1\}} = R_{\mathsf{u},\{1\}} = 0$, $V_1 = U = \emptyset$, and $W = f_1(X, Y)$. Then, the inequalities (4.9), (4.11), (4.12), (4.14) become redundant.

4. Proposition 4.1
   Set $R_{\mathsf{c},\{2\}} = R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = 0$, $U = V_{\mathsf{c}} = V_2 = \emptyset$, and $V_1 = V$. Then, the inequalities (4.7), (4.8), (4.10), (4.11), (4.14) become redundant. Finally, note that the bound on sum rate (4.13) can be relaxed to the bound on the individual private update rate $R_{\mathsf{u},\{2\}}$.

5. Proposition 4.3
   Set $R_{\mathsf{c},\{1,2\}} = R_{\mathsf{u},\{1\}} = R_{\mathsf{u},\{2\}} = 0$ and $V_{\mathsf{c}} = \emptyset$. Then, the inequalities (4.7), (4.8), (4.12), (4.13) become redundant. Alternatively, we can set $V_{\mathsf{c}} = Q$, where Q is a time-sharing random variable independent of (X, Y). Then, we attain a convexified achievable rate region.

Distributed Computing with Successive Refinement

5

In many applications, it is of interest to first acquire a coarse description of the data (a "thumbnail").[1] If promising, one may then choose to download the data at a high resolution. Clearly, in this second stage, it is not necessary to transmit the full high-resolution description — we can exploit the fact that the receiver already has partial knowledge and merely send an update or *refinement.* In information theory, this is known as the *successive refinement* source coding problem. A question of obvious interest is whether we can simultaneously be optimal in both stages: Provide the best possible coarse version in the first stage, given the available rate, yet recover the high-resolution version using a total rate no larger than what would have been needed in the regular compression problem. In particular, this would mean that the coarse description is fully useful for the high-resolution version. Sources for which things work out in this ideal way are called *successively refinable.* While important cases of successively refinable sources have been found, it is also known that general sources are not successively refinable.

In this chapter, we study successive refinement under the paradigm of function computation over networks. We assume that the original data sequences are stored on several, spatially separate terminals. Again, the goal is to first download a coarse version, and then, a refinement. We first consider the special case where one of the sources is completely revealed to the decoders, which reduces to coding for computing with successive refinement. We present a single-letter characterization of the optimal rate region and then provide the necessary and sufficient conditions of successive refinability.

Next, for the general setting, we restrict attention to the special case of *full recovery*: After the second stage, the receiver can recover the full, original source sequences. As for the first stage, the desired coarse description can be an arbitrary element-wise function of the source sequences. For example, when all sources are binary, the coarse stage might consist in recovering the element-wise modulo-2 sum of the source sequences. Our main result is that *all* sources are successively refinable

[1] The material of this chapter has appeared in
C.-Y. Wang, and M. Gastpar, "On distributed successive refinement with lossless recovery," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Honolulu, HI, USA, Jul. 2014.

in sum rate, with respect to any function for the coarse stage, as long as in the second stage, the full source sequences must be recovered.

In the last part of the chapter, we study the *joint source-channel coding* problem of distributed computing with successive refinement. In this case, the encoders no longer merely produce bit streams at certain rates. Rather, they have to transmit codewords across a (noisy) MAC towards the decoder. Again, they do so in two stages: the first stage enables the receiver to recover coarse descriptions, and the second stage, to fully recover the source sequences. In the case of a single source, the solution to this problem follows directly from the corresponding source coding problem. For the distributed case, however, this does not apply, and new methods and tools are required. This is true even if the sources are *independent* of each other. Namely, suppose that the coarse stage requires to recover the *sum* of the original sources. Then, it is well known that the solution to this problem does not follow from the solution to the corresponding source coding problem, see [4]. Hence, not surprisingly, a full characterization of the joint source-channel coding problem of distributed computing with successive refinement appears out of reach. Instead, we characterize a particular class of sources, multiple-access channels, and functions for the first stage for which again, we have perfect successive refinability (assuming that in the second stage, the source sequences must be fully recovered).

5.0.1 Successive Refinement for a Single Source

To set the stage, let us briefly review the case of a single source [43, 44, 45]. Recall that the rate–distortion region $\mathcal{R}(D_1, D_2)$ for successively refining a DMS $\langle X \rangle$ with distortion measures d_1 and d_2 is the set of rate pairs (R_1, R_2) such that

$$\begin{aligned} R_1 &\geq I(X; \hat{X}_1), \\ R_1 + R_2 &\geq I(X; \hat{X}_1, \hat{X}_2) \end{aligned}$$

for some conditional pmf $p_{\hat{X}_1, \hat{X}_2|X}$ such that $\mathbb{E}[d_j(X, \hat{X}_j)] \leq D_j$, $j \in \{1, 2\}$.

The DMS $\langle X \rangle$ is said to be successively refinable with respect to distortion measures (d_1, d_2) if the rate pair

$$(R_1, R_2) = (R_{d_1}(D_1), (R_{d_2}(D_2) - R_{d_1}(D_1))^+)$$

is achievable for all distortion pairs (D_1, D_2), where $R_d(D)$ is the rate–distortion function with distortion measure d for a single description. Under a common distortion measure, i.e., $d_1 = d_2 = d$, Equitz and Cover [44] showed that the source is successively refinable if and only if for all $D_1 \leq D_2$, there exists a conditional pmf $p_{\hat{X}_1, \hat{X}_2|X}$ satisfying the Markov condition $\hat{X}_1 \multimap \hat{X}_2 \multimap X$ such that $p_{\hat{X}_1|X}$ and $p_{\hat{X}_2|X}$ attain the rate–distortion function $R_d(D_1)$ and $R_d(D_2)$, respectively.

5.1 Problem Statement

A DMS $\langle S, T \rangle$ generates i.i.d. source sequences (S^k, T^k). We consider two scenarios. In both scenarios, there are two encoding terminals and two decoding terminals. The two encoding terminals observe the source sequences S^k and T^k, respectively. Each decoding terminal $j \in \{1, 2\}$ wishes to recover an element-wise function f_j of the two source sequences. Denote by $\hat{w}_j^k$ the estimate at the j-th decoding terminal. Now we provide the individual details of the two scenarios.

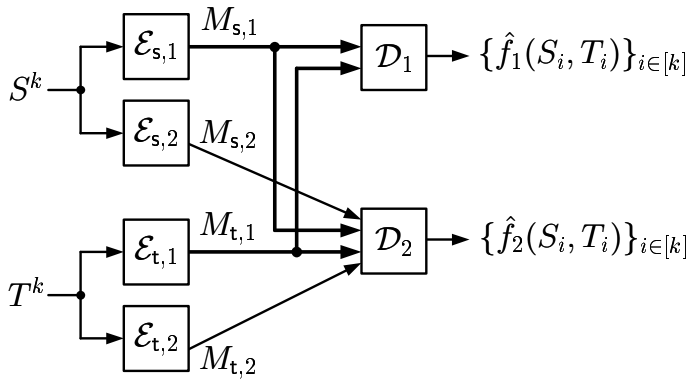

Figure 5.1: The source coding problem of distributed computing with successive refinement.

5.1.1 Distributed Source Coding

Consider the system depicted in Figure 5.1. Each encoding terminal (indexed by $j \in \{\mathsf{s}, \mathsf{t}\}$) generates two descriptions $M_{j,1}, M_{j,2}$ of rates $R_{j,1}, R_{j,2}$, respectively. The first decoding terminal only receives the descriptions $(M_{\mathsf{s},1}, M_{\mathsf{t},1})$ and the second decoding terminal receives all four descriptions $(M_{\mathsf{s},1}, M_{\mathsf{t},1}, M_{\mathsf{s},2}, M_{\mathsf{t},2})$.

A $(2^{kR_{\mathsf{s},1}}, 2^{kR_{\mathsf{t},1}}, 2^{kR_{\mathsf{s},2}}, 2^{kR_{\mathsf{t},2}}, k)$ distributed multiple description code consists of

- four encoders, where Encoder (s, j) $(j \in \{1, 2\})$ assigns an index $m_{\mathsf{s},j}(s^k) \in [2^{kR_{\mathsf{s},j}}]$ to each sequence $s^k \in \mathcal{S}^k$ and Encoder (t, j) $(j \in \{1, 2\})$ assigns an index $m_{\mathsf{t},j}(t^k) \in [2^{kR_{\mathsf{t},j}}]$ to each sequence $t^k \in \mathcal{T}^k$, and
- two decoders, where Decoder 1 assigns an estimate $\hat{w}_1^k$ to each index pair $(m_{\mathsf{s},1}, m_{\mathsf{t},1})$ and Decoder 2 assigns an estimate $\hat{w}_2^k$ to each index quadruple $(m_{\mathsf{s},1}, m_{\mathsf{t},1}, m_{\mathsf{s},2}, m_{\mathsf{t},2})$.

A rate quadruple $(R_{\mathsf{s},1}, R_{\mathsf{t},1}, R_{\mathsf{s},2}, R_{\mathsf{t},2})$ is said to be achievable if there exists a sequence of $(2^{kR_{\mathsf{s},1}}, 2^{kR_{\mathsf{t},1}}, 2^{kR_{\mathsf{s},2}}, 2^{kR_{\mathsf{t},2}}, k)$ codes with

$$\text{(C1)} \quad \lim_{k\to\infty} \mathbb{P}\left(\bigcup_{i=1}^{k} \left\{\hat{W}_{1i} \neq f_1(S_i, T_i)\right\}\right) = 0;$$

$$\text{(C2)} \quad \lim_{k\to\infty} \mathbb{P}\left(\bigcup_{i=1}^{k} \left\{\hat{W}_{2i} \neq f_2(S_i, T_i)\right\}\right) = 0.$$

The optimal rate region $\mathcal{R}_{\mathsf{SuccRef}}$ is the closure of the set of achievable rate quadruples. Let us also introduce the other rate regions of interest. Denote by $\mathcal{R}_{\mathrm{C}j}$ the optimal rate region when only (Cj) has to be satisfied, $j \in \{1, 2\}$.

5.1.2 Joint Source–Channel Coding

Consider the two-hop network depicted in Figure 5.2. In addition to the encoding and decoding terminals, there is a node in between serving as a passive router. The channel between the encoding terminals and the router is a 2-sender discrete

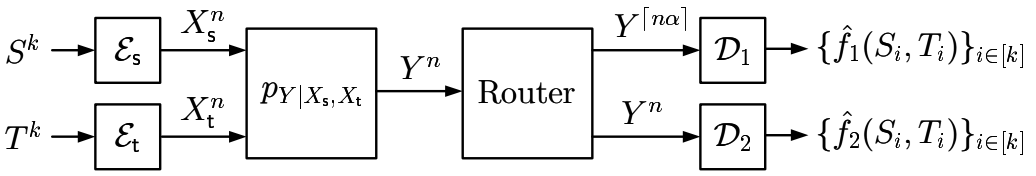

Figure 5.2: The joint source–channel coding problem of distributed computing with successive refinement.

memoryless MAC $\langle p_{Y|X_{\mathsf{s}},X_{\mathsf{t}}} \rangle$. The channels between the router and the decoding terminals are noiseless links with unlimited capacity. The router passively relays all received signals to the second decoding terminal but only a fraction of them to the first decoding terminal.

Let $\alpha \in [0,1]$. A $(|\mathcal{S}|^k, |\mathcal{T}|^k, \alpha, n)$ joint source–channel code consists of

- two encoders, where Encoder s assigns a sequence $x_{\mathsf{s}}^n(s^k) \in \mathcal{X}_{\mathsf{s}}^n$ to each sequence $s^k \in \mathcal{S}^k$ and Encoder t assigns a sequence $x_{\mathsf{t}}^n(t^k) \in \mathcal{X}_{\mathsf{t}}^n$ to each sequence $t^k \in \mathcal{T}^k$, and
- two decoders, where Decoder 1 assigns an estimate $\hat{w}_1^k$ to each sequence $y^{\lceil \alpha n \rceil} \in \mathcal{Y}^{\lceil \alpha n \rceil}$ and Decoder 2 assigns an estimate $\hat{w}_2^k$ to each sequence $y^n \in \mathcal{Y}^n$.

Define the rates $R_1 := \lceil \alpha n \rceil / k$ and $R_2 := n/k$ as the number of channel uses per letter. We say that a rate pair (R_1, R_2) is achievable if there exists a sequence of $(|\mathcal{S}|^k, |\mathcal{T}|^k, \alpha, n)$ codes satisfying both (C1) and (C2). The optimal rate region $\mathcal{R}_{\mathsf{SuccRef}}$ is the closure of the set of achievable rate pairs.[2] Other rate regions of interest are defined similarly as in Section 5.1.1. Furthermore, we define

$$R_1^* := \min\{R_1 \big| (R_1, R_2) \in \mathcal{R}_{\mathrm{C1}}\},$$
$$R_2^* := \min\{R_2 \big| (R_1, R_2) \in \mathcal{R}_{\mathrm{C2}}\}.$$

5.2 Coding for Computing with Successive Refinement

Before going directly to the general distributed setting, let us first consider the extreme case where $R_{\mathsf{t},1} > H(T)$, i.e., the two decoders can learn the whole sequence T^k losslessly. This case can be interpreted as an extension of the successive refinement problem to include common side information at decoders. We remark that successive refinement with distinct side information has been considered in [46, 47]. Here our emphasis is placed on the lossless reconstruction of element-wise functions of source and side information, instead of a lossy reconstruction of the source itself. Let us define

$$\mathcal{R}_{\mathsf{s}|\mathsf{t}}^* = \{(R_{\mathsf{s},1}, R_{\mathsf{s},2}) : (R_{\mathsf{s},1}, H(T), R_{\mathsf{s},2}, 0) \in \mathcal{R}^*\}.$$

As can be seen from Figure 5.3, this case corresponds to Configuration $(\mathcal{A}_{\mathsf{c}}|\mathcal{A}_{\mathsf{s}}) = (\{2\}, \{1,2\}|\emptyset)$ in the problem of sequential coding for computing. Thus, we have the following theorem, which is a simple consequence of Theorem 4.3.

[2] For convenience, in both scenarios we use the same set of notations to refer to the rate regions.

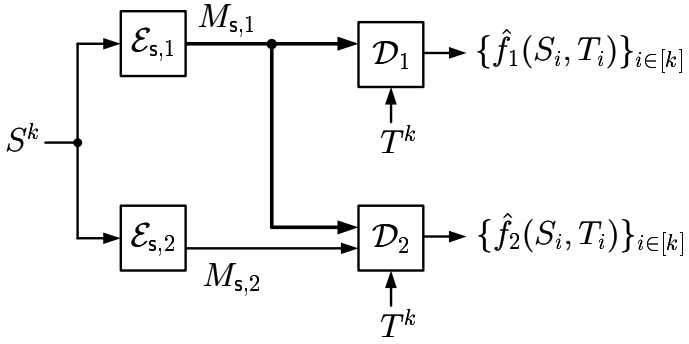

Figure 5.3: Coding for Computing with Successive refinement.

Theorem 5.1. *The rate region $\mathcal{R}^*_{\mathsf{s}|\mathsf{t}}$ is the set of rate pairs $(R_{\mathsf{s},1}, R_{\mathsf{s},2})$ such that*

$$\begin{aligned} R_{\mathsf{s},1} &\geq I(S; V_1|T), \\ R_{\mathsf{s},1} + R_{\mathsf{s},2} &\geq I(S; V_1, V_2|T), \end{aligned}$$

for some conditional pmf $p_{V_1,V_2|S}$ with $|\mathcal{V}_1| \leq |\mathcal{S}| + 2$ and $|\mathcal{V}_2| \leq |\mathcal{S}||\mathcal{V}_1| + 1$ such that $H(f_j(S,T)|V_j, T) = 0$, $j \in \{1, 2\}$.

We say that the DMS $\langle S, T\rangle$ is successively refinable with respect to the functions (f_1, f_2) if the rate pair

$$(R_{\mathsf{s},1}, R_{\mathsf{s},2}) = (R^*_{f_1}, (R^*_{f_2} - R^*_{f_1})^+)$$

is achievable, where R^*_f is the optimal compression rate for losslessly computing a single function $f(s,t)$. Similar to the original setup without side information, we have the following two necessary and sufficient conditions of successive refinability depending on whether $R^*_{f_2} - R^*_{f_1}$ is nonnegative or nonpositive.

Proposition 5.1. *Assume that $R^*_{f_2} \geq R^*_{f_1}$. Then, the DMS $\langle S, T\rangle$ is successively refinable with respect to the functions (f_1, f_2) if and only if there exists a conditional pmf $p_{V_1,V_2|S}$ satisfying the Markov condition $V_1 \multimap (V_2, T) \multimap S$ such that $p_{V_1|S}$ and $p_{V_2|S}$ attain the optimal compression rates $R^*_{f_1}$ and $R^*_{f_2}$, respectively.*

Proof: First, note that in this case $(R^*_{f_2} - R^*_{f_1})^+ = R^*_{f_2} - R^*_{f_1}$.

(Sufficiency) Assume that the conditional pmf $p_{V_1,V_2|X}$ satisfies the mentioned condition. Then, we have

$$\begin{aligned} R_{\mathsf{s},1} &\geq I(S; V_1|T) = R^*_{f_1}, \\ R_{\mathsf{s},1} + R_{\mathsf{s},2} &\geq I(S; V_1, V_2|T) \\ &\overset{(a)}{=} I(S; V_2|T) = R^*_{f_2}, \end{aligned}$$

where (a) follows from the Markov condition $V_1 \multimap (V_2, T) \multimap S$. Thus, the rate pair $(R_{\mathsf{s},1}, R_{\mathsf{s},2}) = (R^*_{f_1}, R^*_{f_2} - R^*_{f_1})$ is achievable.

(Necessity) Assume that the DMS $\langle S, T\rangle$ is successively refinable with respect to the functions (f_1, f_2). Then, it requires that there exists a conditional pmf $p_{V_1,V_2|S}$

satisfying

$$\begin{aligned} R^*_{f_1} &\geq I(S;V_1|T), \\ R^*_{f_2} &\geq I(S;V_1,V_2|T) \geq I(S;V_2|T), \\ H(f_j(S,T)|V_j,T) &= 0, j \in \{1,2\}. \end{aligned}$$

Since the last condition implies that $I(S;V_j|T) \geq R^*_{f_j}$, $j \in \{1,2\}$, we must have

$$\begin{aligned} I(S;V_1|T) &= R^*_{f_1}, \\ I(S;V_2|T) &= R^*_{f_2}, \\ I(S;V_1|V_2,T) &= 0. \end{aligned}$$

■

We remark that Proposition 5.1 implies that all DMSs $\langle S,T\rangle$ are successively refinable with respect to any function f_1 in the first stage as long as $f_2(s,t) = (s,t)$.

Proposition 5.2. *Assume that $R^*_{f_2} \leq R^*_{f_1}$. Then, the DMS $\langle S,T\rangle$ is successively refinable with respect to the functions (f_1,f_2) if and only if there exists a conditional pmf $p_{V_1|S}$ that satisfies $H(f_2(S,T)|V_1,T) = 0$ and attains the optimal compression rate $R^*_{f_1}$.*

Proof: First, note that in this case $(R^*_{f_2} - R^*_{f_1})^+ = 0$.

(Sufficiency) Assume that the conditional pmf $p_{V_1|S}$ satisfies the mentioned condition. Then, by setting $V_2 = V_1$ in Theorem 5.1, we have

$$\begin{aligned} R_{\mathsf{s},1} &\geq I(S;V_1|T) = R^*_{f_1}, \\ R_{\mathsf{s},1} + R_{\mathsf{s},2} &\geq I(S;V_1|T) = R^*_{f_1}, \end{aligned}$$

Thus, the rate pair $(R_{\mathsf{s},1}, R_{\mathsf{s},2}) = (R^*_{f_1}, 0)$ is achievable.

(Necessity) Assume that the DMS $\langle S,T\rangle$ is successively refinable with respect to the functions (f_1,f_2). Then, it requires that there exists a conditional pmf $p_{V_1,V_2|S}$ satisfying

$$\begin{aligned} R^*_{f_1} &\geq I(S;V_1|T), \\ R^*_{f_1} &\geq I(S;V_1,V_2|T) \geq I(S;V_1|T), \\ H(f_j(S,T)|V_j,T) &= 0, j \in \{1,2\}. \end{aligned}$$

Since the last condition implies that $I(S;V_1|T) \geq R^*_{f_1}$, we must have

$$\begin{aligned} I(S;V_1|T) &= R^*_{f_1}, \\ I(S;V_2|V_1,T) &= 0. \end{aligned}$$

Finally, since the conditions $H(f_2(S,T)|V_2,T) = 0$ and $I(S;V_2|V_1,T) = 0$ imply that $H(f_2(S,T)|V_1,T) = 0$, the proof is complete. ■

If $R^*_{f_2} \leq R^*_{f_1}$, the condition $H(f_2(S,T)|V_1,T) = 0$ implies that in order to be successively refinable, one should be able to recover both functions already in the first stage. Namely, the second stage becomes degenerated.

5.3 Distributed Source Coding

In this section, we consider the source coding problem of distributed computing with successive refinement and restrict attention to full recovery in the second stage, i.e., $f_2(s,t) = (s,t)$. We are mainly interested in *successive refinability*, i.e., under what condition it holds that $\mathcal{R}_{\mathsf{SuccRef}} = \mathcal{R}_{\mathrm{C1}} \cap \mathcal{R}_{\mathrm{C2}}$. A direct approach for solving the successive refinability problem would be to completely characterize the optimal rate region $\mathcal{R}_{\mathsf{SuccRef}}$. Unfortunately, the problem setup includes distributed (lossless) coding for computing as a special case, which remains open in general. Thus, we adopt an indirect approach instead.

Proposition 5.1 in the last section implies that if the source T is known at both decoders, then all DMSs $\langle S,T\rangle$ are successively refinable with respect to any function f_1 in the first stage as long as $f_2(s,t) = (s,t)$. One may wonder whether the same holds for the distributed setting. The following simple example shows that it is in general not the case. For convenience, in this section f_1 is simply denoted by f.

Example 5.1. *Let $S = (A,B)$ and $T = A$, where A, B are i.i.d. Bernoulli(1/2). Assume that $f(s,t) = t$. Then, it can be checked that $(R_{\mathsf{s},1}, R_{\mathsf{t},1}, R_{\mathsf{s},2}, R_{\mathsf{t},2}) = (1,0,0,1) \in \mathcal{R}_{C1} \cap \mathcal{R}_{C2}$. Indeed, to satisfy (C1), it suffices that Encoder $(\mathsf{s},1)$ directly sends a^k without coding. In order to satisfy (C2), it suffices that Encoder $(\mathsf{s},1)$ sends b^k without coding and Encoder $(\mathsf{t},2)$ sends a^k without coding.*

However, $(1,0,0,1) \notin \mathcal{R}_{\mathsf{SuccRef}}$. The reason is that such rate requirement results in the following dilemma: Encoder $(\mathsf{s},1)$ must communicate a^k to Decoder 1 to fulfill (C1) and at the same time communicate b^k to Decoder 2 to fulfill (C2) because it is the only encoder observing b^k.

Then, how about sum rate? Let us define the *sum-rate rate losses* for the first and second stage, respectively, as

$$\begin{aligned}\Delta_1 &:= R_{\mathsf{s},1} + R_{\mathsf{t},1} - R^*(f),\\ \Delta_2 &:= R_{\mathsf{s},1} + R_{\mathsf{t},1} + R_{\mathsf{s},2} + R_{\mathsf{t},2} - H(S,T).\end{aligned}$$

where

$$R^*(f) := \min\{R_{\mathsf{s},1} + R_{\mathsf{t},1} \big| (R_{\mathsf{s},1}, R_{\mathsf{t},1}, R_{\mathsf{s},2}, R_{\mathsf{t},2}) \in \mathcal{R}_{\mathrm{C1}}\}.$$

Definition 5.1 (Successive Refinability in Sum Rate)**.** *The DMS $\langle S,T\rangle$ is successively refinable in sum rate with respect to f if there exists a rate quadruple in $\mathcal{R}_{\mathsf{SuccRef}}$ satisfying $\Delta_1 = \Delta_2 = 0$.*

We now present the main theorem in this chapter, which says that both decoding terminals can attain their individual optimal sum rates.

Theorem 5.2. *Consider the source coding problem of distributed computing with successive refinement. If full recovery in the second stage is required, then all DMSs $\langle S,T\rangle$ are successively refinable in sum rate with respect to any function f.*

Proof: Given *any* code for the first (coarse) stage of our problem, we first construct an improved version of that code by removing possible redundancies. Then, we construct a code for the second stage and show that for our overall code, we have

$\Delta_2 = 0$. However, since the code of the first stage was arbitrary, the argument in particular also applies to the optimal code for the first stage, i.e., the code for which (by definition) $\Delta_1 = 0$ (though, as pointed out, for most instances of the problem, no efficient explicit description of this code is known). This proves the theorem.

Stage 1: More explicitly, to remove possible redundancies from the first-stage code, we construct an extended "super-letter" code, as follows: Run Encoder $(\mathsf{s}, 1)$ and $(\mathsf{t}, 1)$ for B blocks of k source letters to output the sequences of descriptions $M_{\mathsf{u},1}^B$ and $M_{\mathsf{v},1}^B$. Then, we treat $(M_{\mathsf{s},1}, M_{\mathsf{t},1})$ as super letters and apply Slepian–Wolf coding on $(M_{\mathsf{s},1}, M_{\mathsf{t},1})$.

Both decoding terminals can recover the super letters with vanishing error as B increases if

$$\begin{aligned} R_{\mathsf{s},1} &> \frac{1}{k} H(M_{\mathsf{s},1}|M_{\mathsf{t},1}), \\ R_{\mathsf{t},1} &> \frac{1}{k} H(M_{\mathsf{t},1}|M_{\mathsf{s},1}), \\ R_{\mathsf{s},1} + R_{\mathsf{t},1} &> \frac{1}{k} H(M_{\mathsf{s},1}, M_{\mathsf{t},1}). \end{aligned} \tag{5.1}$$

Finally, at the first decoding terminal, we run Decoder 1 on each pair of the estimated descriptions $(\hat{m}_{\mathsf{s},1}, \hat{m}_{\mathsf{t},1})$ to get the estimates $\hat{w}_1^k$. Hence, this improved code has rates no larger than the rates of the original code and enables the same reconstruction quality.

Stage 2 (refinement): If we treat (S^k, T^k) as super letters, then Decoder 2 can use $(M_{\mathsf{s},1}, M_{\mathsf{t},1})$ recovered in the first stage as side information. For the refinement, we apply Slepian–Wolf coding on (S^k, T^k) assuming decoder side information $(M_{\mathsf{s},1}, M_{\mathsf{t},1})$.

At the second decoding terminal, the super letters (S^k, T^k) can be recovered with vanishing error as the number of blocks B increases if

$$\begin{aligned} R_{\mathsf{s},2} &> \frac{1}{k} H(S^k|T^k, M_{\mathsf{s},1}, M_{\mathsf{t},1}), \\ R_{\mathsf{t},2} &> \frac{1}{k} H(T^k|S^k, M_{\mathsf{s},1}, M_{\mathsf{t},1}), \\ R_{\mathsf{s},2} + R_{\mathsf{t},2} &> \frac{1}{k} H(S^k, T^k|M_{\mathsf{s},1}, M_{\mathsf{t},1}). \end{aligned} \tag{5.2}$$

Thus, Expressions (5.1) and (5.2) imply that the sum-rate rate loss Δ_2 can be driven to zero as closely as desired by increasing B. We remark that even if k remains fixed, $\Delta_2 = 0$ is still achievable. ■

In the proof of Theorem 5.2, we have used the so-called "super-letter" argument, in which we use an existing code as a module to build a larger block code. In some cases, it suffices to simply *append* additional components to the existing code. The following is an example for which we have a simple solution to successive refinement.

Example 5.2. *Let $S \sim$ Bernoulli(1/2) and $T = S \oplus Z$ for some $Z \sim$ Bernoulli(q) independent of S, where $q \in (0, 1/2]$. We assume that the first decoding terminal wishes to recover $S \oplus T (= Z)$. In this case, Körner–Marton coding achieves the optimal sum rate $2H(Z)$ [3]. To perform successive refinement in the second stage, we use random linear binning again. Denote by $\mathbf{H}_1$ the compression matrix used in the first stage, which is of size $kH(Z) \times k$. In the second stage, we fix a rate*

pair $(R_{\mathsf{s},2}, R_{\mathsf{t},2})$ *satisfying* $R_{\mathsf{s},2} + R_{\mathsf{t},2} = 1 - H(Z)$ *(and the appropriate side rate constraints). Then, we generate a matrix* $\mathbf{H}_2$ *of size* $k \max\{R_{\mathsf{s},2}, R_{\mathsf{t},2}\} \times k$ *with i.i.d. Bernoulli*$(1/2)$ *entries. Denote* $\mathbf{H} = \begin{bmatrix}\mathbf{H}_1^T & \mathbf{H}_2^T\end{bmatrix}^T$*. To the best of our knowledge, in all achievability proofs of Slepian–Wolf rate region via random (linear) binning, the codebooks are generated independently. However, as demonstrated in Appendix, under the assumption that* $\mathbb{P}(S \neq T) > 0$*, it suffices to use* $\mathbf{H}$ *at one encoder and a submatrix of* $\mathbf{H}$ *at the other.*[3] *Therefore, a lossless recovery of the full source sequences can be achieved and* $\Delta_2 = 0$ *is achievable in a simple manner.*

In fact, the above example satisfies that $\mathcal{R}_{\mathsf{SuccRef}} = \mathcal{R}_{\mathrm{C1}} \cap \mathcal{R}_{\mathrm{C2}}$. The optimal sum rate of $(C1)$ is achieved at only one point $(H(Z), H(Z))$, which allows us to move to any point achieving the optimal sum rate of (C2). On the other hand, if Decoder 2 only uses the recovered Z^k as decoder side information, then it requires $R_{\mathsf{s},2} + R_{\mathsf{t},2} \geq H(S,T|Z)$ to recover the entire source sequences. The sum-rate rate loss becomes

$$\Delta_2 = 2H(Z) + H(S,T|Z) - H(S,T) = H(Z).$$

Therefore, in general, in order to attain $\Delta_2 = 0$, Decoder 2 must use both descriptions $M_{\mathsf{s},1}, M_{\mathsf{t},1}$ as side information.

5.4 Joint Source–Channel Coding

Let us turn to joint source–channel coding. In this section, we assume that the sources are independent and again restrict attention to full recovery in the second stage, i.e., $f_2(s,t) = (s,t)$. For convenience, in this section f_1 is simply denoted by f. We are mainly interested in successive refinability.

Definition 5.2 (Successive Refinability)**.** *The DMS* $\langle S,T\rangle$ *is successively refinable over the MAC* $\langle p_{Y|X_{\mathsf{s}},X_{\mathsf{t}}}\rangle$ *with respect to* f *if* $(R_1^*, R_2^*) \in \mathcal{R}_{\mathsf{SuccRef}}$*.*

If only (C1) is demanded, the problem reduces to computation over MAC but R_1^* is not known in general. Thus, $\mathcal{R}_{\mathsf{SuccRef}}$ is also not known in general and we need to address the successive refinability using an indirect approach. We remark that if only (C2) is demanded, the problem reduces to communication of independent sources over MAC and R_2^* is known [5, Example 14.2].

Let us define

$$R_{\mathsf{refine}}^* := \min\{R_2 \big| (R_1^*, R_2) \in \mathcal{R}_{\mathsf{SuccRef}}\} - R_1^*,$$

and then we have

$$R_1^* + R_{\mathsf{refine}}^* \geq R_2^*. \tag{5.3}$$

Thus, an equivalent condition of successive refinability is that the equality in (5.3) holds. An operational meaning for R_{refine}^* is as follows. Assume that in the first kR_1^* time slots, we use an optimal code targeted on (C1). Then, kR_{refine}^* is the minimum time slots required to recover all source letters at the second decoding terminal.

[3] We remark that the same construction has been used in [48].

Again, we start with an example showing that even if we ask full recovery in the second stage, a DMS can still fail to be successively refinable with respect to some function f.

Example 5.3. *Let S and T be i.i.d. Bernoulli(1/2). Consider the following arithmetic adder MAC:*

$$Y = X_{\mathsf{s}} + X_{\mathsf{t}},$$

where $X_{\mathsf{s}}, X_{\mathsf{t}} \in \{0,1\}$ and $Y \in \{0,1,2\}$. Assume that the first decoding terminal simply wishes to recover S losslessly, i.e., $f(s,t) = s$. We have

$$R_1^* = \frac{H(S)}{\max\limits_{p_{X_{\mathsf{s}}}, x_{\mathsf{t}}} I(X_{\mathsf{s}}; Y | X_{\mathsf{t}} = x_{\mathsf{t}})} = 1,$$
$$R_2^* = \frac{H(S) + H(T)}{\max\limits_{p_{X_{\mathsf{s}}}, p_{X_{\mathsf{t}}}} I(X_{\mathsf{s}}, X_{\mathsf{t}}; Y)} = 4/3.$$

From the perspective of the first decoding terminal, any information about T sent in the first stage is interference since S and T are independent. In order to achieve R_1^*, the second decoding terminal cannot learn any information about T in the first stage and we have $R_{\mathsf{refine}}^* = 1$. Therefore, in this case the inequality in (5.3) *is strict.*

Next, we provide an upper bound on R_{refine}^* and consider two special cases in the following subsections.

Proposition 5.3.

$$R_{\mathsf{refine}}^* \le \min_{p_Q p_{X_{\mathsf{s}}|Q} p_{X_{\mathsf{t}}|Q}} \max \left\{ \frac{H(S|T, f(S,T))}{I(X_{\mathsf{s}}; Y | X_{\mathsf{t}}, Q)}, \frac{H(T|S, f(S,T))}{I(X_{\mathsf{t}}; Y | X_{\mathsf{s}}, Q)}, \frac{H(S,T|f(S,T))}{I(X_{\mathsf{s}}, X_{\mathsf{t}}; Y | Q)} \right\}, \tag{5.4}$$

where $|\mathcal{Q}| \le 4$.

Proof: After the first stage, the second decoding terminal can also recover $\{f(s_i, t_i)\}_{i \in [k]}$ and then use them as side information. The rest follows from [49, Theorem 5.3] immediately. ∎

5.4.1 Computing Linear functions over Linear and Symmetric MACs

Consider any finite field $\mathbb{F}$. Let $f(s,t)$ be an $\mathbb{F}$-linear function and let the MAC $\langle p_{Y|X_{\mathsf{s}}, X_{\mathsf{t}}} \rangle$ be $\mathbb{F}$-linear and symmetric (see Section 2.7). Then, we have the following proposition.

Proposition 5.4. *Consider the joint source–channel coding problem of distributed computing with successive refinement. If full recovery in the second stage is required, then all DMSs $\langle S, T \rangle$ are successively refinable over any $\mathbb{F}$-linear and symmetric MAC $\langle p_{Y|X_{\mathsf{s}}, X_{\mathsf{t}}} \rangle$ with respect to any $\mathbb{F}$-linear function f.*

Proof: From [4, Theorem 1], we have

$$R_1^* = \frac{H(f(S,T))}{C},$$

where $C := \max_{p_{X_\mathsf{s}} p_{X_\mathsf{t}}} I(X_\mathsf{s}, X_\mathsf{t}; Y)$ can be achieved by the uniform distribution. Also, it is easy to show that $R_2^* = \frac{H(S,T)}{C}$. By setting $\mathcal{Q} = \emptyset$ and p_{X_s}, p_{X_t} uniform in Proposition 5.3, we have the upper bound

$$R_{\mathsf{refine}}^* \leq \frac{H(S,T|f(S,T))}{C}.$$

Finally, it is a simple task to establish that Expression (5.3) holds with equality. ■

Particularly, Proposition 5.4 implies that if in Example 5.3 the arithmetic adder MAC is replaced by a modulo-2 adder MAC, i.e., $Y = X_\mathsf{s} \oplus X_\mathsf{t}$, the DMS becomes successively refinable.

5.4.2 Computing Partially Invertible Functions of Sources with Equal Entropy

A function $f(s,t)$ is said to be partially invertible with respect to s if s can be deduced from $f(s,t)$ and t. If $H(S) = H(T)$ and $f(s,t)$ is partially invertible with respect to both s and t, then we have the following sufficient condition of successive refinability.

Proposition 5.5. *Consider the joint source–channel coding problem of distributed computing with successive refinement. Assume full recovery in the second stage. Furthermore, assume that $H(S) = H(T)$ and that $f(s,t)$ is partially invertible with respect to both s and t. Then, the DMS $\langle S, T\rangle$ is successively refinable over the MAC $\langle p_{Y|X_\mathsf{s},X_\mathsf{t}}\rangle$ with respect to f if*

$$R_1^* = \frac{H(f(S,T))}{\max\limits_{p_{X_\mathsf{s}} p_{X_\mathsf{t}}} I(X_\mathsf{s}, X_\mathsf{t}; Y)}. \tag{5.5}$$

Proof: Since f is partially invertible with respect to both s and t, we have $H(S|T, f(S,T)) = H(T|S, f(S,T)) = 0$. Thus, Expression (5.4) can be simplified as

$$R_{\mathsf{refine}}^* \leq \frac{H(S,T|f(S,T))}{\max\limits_{p_{X_\mathsf{s}} p_{X_\mathsf{t}}} I(X_\mathsf{s}, X_\mathsf{t}; Y)}. \tag{5.6}$$

On the other hand, since $H(S,T) = H(S) + H(T) = 2H(S)$, it can be easily shown that

$$R_2^* = \frac{H(S,T)}{\max\limits_{p_{X_\mathsf{s}} p_{X_\mathsf{t}}} I(X_\mathsf{s}, X_\mathsf{t}; Y)}. \tag{5.7}$$

Thus, if (5.5) holds, combining with (5.6) and (5.7) shows that (5.3) holds with equality and the proposition is established. ■

Finally, we provide an example which shows that the upper bound (5.4) of R_{refine}^* is in general loose and thus Condition (5.5) is not a necessary condition.

Example 5.4. *Let S and T be i.i.d. Bernoulli$(1/2)$. Consider the following deterministic MAC:*

$$\begin{aligned} X_\mathsf{s} &= (X_{\mathsf{s},1}, X_{\mathsf{s},2}), \\ Y &= (X_{\mathsf{s},1} \oplus X_\mathsf{t}, X_{\mathsf{s},2}), \end{aligned}$$

where $X_{\mathsf{s},1}, X_{\mathsf{s},2}, X_{\mathsf{t}} \in \{0,1\}$. *The desired function at the first decoding terminal is* $f(s,t) = s \oplus t$.

Since $H(S) = H(T)$, *we have* $R_2^* = 1$ *from Expression* (5.7). *Also, we have the lower bound*

$$R_1^* \geq \frac{H(f(S,T)|S)}{\max\limits_{p_{X_{\mathsf{s}}} p_{X_{\mathsf{t}}}} I(X_{\mathsf{t}}; Y|X_{\mathsf{s}})} = 1.$$

Since $R_2^* \geq R_1^*$, *we conclude that* $R_1^* = 1$. *Then, since* $R_1^* = R_2^*$, *it means that the entire source sequences can already be learned in the first stage and the sources are trivially successively refinable. Finally, we have* $R_{\mathsf{refine}}^* = 0 < 1/2$, *so the upper bound* (5.4) *is loose.*

The above example also shows that it is possible that R_{refine}^* is not achievable, even if a code attaining R_1^* is used in the first stage. Consider $X_{\mathsf{s}} = (S, 0)$ and $X_{\mathsf{t}} = T$. With this code, only the desired function can be learned in the first stage and it is clear that $R_{\mathsf{refine}}^* = 0$ is not achievable.

Appendix: Dependent Codebooks

The proof is similar to [5, Section 10.3], so we just point out the difference. Denote by $R_{\mathsf{s}}, R_{\mathsf{t}}$ the compression rates of Encoders s and t, respectively. Without loss of generality, we assume that $R_{\mathsf{s}} \geq R_{\mathsf{t}}$. The compression matrix $\mathbf{H}_{\mathsf{s}}$ of size $kR_{\mathsf{s}} \times k$ is generated i.i.d. Bernoulli(1/2). Denote by $\mathbf{H}_{\mathsf{t}}$ the first kR_{t} rows of $\mathbf{H}_{\mathsf{s}}$. The decoder uses joint typicality decoding.

We now analyze the probability of error. Fix $\epsilon \in (0,1)$. Denote by M_{s} and M_{t} the random bin indices of S^k and T^k, respectively. The decoder makes an error if and only if one or more of the following events occur:

$$\begin{aligned}
&\mathcal{E}_1 = \{(S^k, T^k) \notin \mathcal{T}_\epsilon^{(k)}\},\\
&\mathcal{E}_2 = \{\hat{s}^k \in \mathcal{B}_{\mathsf{s}}(M_{\mathsf{s}}) \text{ for some } \hat{s}^k \neq S^k,\, (\hat{s}^k, T^k) \in \mathcal{T}_\epsilon^{(k)}\},\\
&\mathcal{E}_3 = \{\hat{t}^k \in \mathcal{B}_{\mathsf{t}}(M_{\mathsf{t}}) \text{ for some } \hat{t}^k \neq T^k,\, (S^k, \hat{t}^k) \in \mathcal{T}_\epsilon^{(k)}\},\\
&\mathcal{E}_4 = \{\hat{s}^k \in \mathcal{B}_{\mathsf{s}}(M_{\mathsf{s}}), \hat{t}^k \in \mathcal{B}_{\mathsf{t}}(M_{\mathsf{t}}) \text{ for some } \hat{s}^k \neq S^k, \hat{t}^k \neq T^k,\, (\hat{s}^k, \hat{t}^k) \in \mathcal{T}_\epsilon^{(k)}\},
\end{aligned}$$

where $\mathcal{B}_j(m)$ is the set of sequences which are assigned the bin index m. Clearly, using the same compression matrix does not affect the first three error events. Thus, it suffices to check the last error event $\mathcal{E}_4$. It is straightforward to show that

$$\mathbb{P}(\mathcal{E}_4) \leq \sum_{s^k, t^k} \mathbb{P}\left(S^k = s^k, T^k = t^k\right) \sum_{\substack{(\hat{s}^k, \hat{t}^k) \in \mathcal{T}_\epsilon^{(k)} \\ \hat{s}^k \neq s^k,\, \hat{t}^k \neq t^k}} \mathbb{P}\left(\hat{s}^k \in \mathcal{B}_{\mathsf{s}}(\mathbf{H}_{\mathsf{s}} s^k), \hat{t}^k \in \mathcal{B}_{\mathsf{t}}(\mathbf{H}_{\mathsf{t}} t^k)\right).$$

The assumption $\mathbb{P}(S \neq T) > 0$ implies that there exists $(s', t') \in \mathcal{S} \times \mathcal{T}$ such that $s' \neq t'$ and $p_{S,T}(s', t') > 0$. Thus, the joint empirical pmf of any two identical sequences $(\hat{s}^k, \hat{s}^k)$ evaluated at (s', t') is zero. With $\epsilon < 1$, two identical sequences cannot be jointly typical, i.e., $\hat{s}^k \neq \hat{t}^k$ for all $(\hat{s}^k, \hat{t}^k) \in \mathcal{T}_\epsilon^{(k)}$. If additionally $\hat{s}^k \neq s^k$ and $\hat{t}^k \neq t^k$, the two events $\{\hat{s}^k \in \mathcal{B}_{\mathsf{s}}(\mathbf{H}_{\mathsf{s}} s^k)\}$ and $\{\hat{t}^k \in \mathcal{B}_{\mathsf{t}}(\mathbf{H}_{\mathsf{t}} t^k)\}$ are independent

and thus

$$\begin{aligned}\mathbb{P}(\mathcal{E}_4) \leq & \sum_{\substack{(\hat{s}^k,\hat{t}^k)\in\mathcal{T}_\epsilon^{(k)} \\ \hat{s}^k\neq s^k,\, \hat{t}^k\neq t^k}} \mathbb{P}\left(\hat{s}^k \in \mathcal{B}_{\mathsf{s}}(\mathbf{H}_{\mathsf{s}} s^k)\right) \mathbb{P}\left(\hat{t}^k \in \mathcal{B}_{\mathsf{t}}(\mathbf{H}_{\mathsf{t}} t^k)\right) \\ & \leq \; 2^{k(H(S,T)+\delta(\epsilon))} 2^{-k(R_{\mathsf{s}}+R_{\mathsf{t}})}.\end{aligned}$$

Therefore, $\mathbb{P}(\mathcal{E}_4) \to 0$ as $k \to \infty$ if $R_{\mathsf{s}} + R_{\mathsf{t}} > H(S,T) + \delta(\epsilon)$.

Computation over Linear Multiple Access Channels

6

To date, wireless sensor networks have been deployed for various applications in environmental monitoring, e.g., air/water quality monitoring and forest fire detection.[1] Typically, a sensor network consists of a single fusion center and multiple sensors measuring certain parameters. Sensor deployment can be costly, so the lifetime of sensors is expected to be months or even years. Therefore, *power efficiency* becomes an important issue for system design.

Traditionally, sensors simply convey all the measured parameters to the fusion center. However, for many applications, the fusion center is only interested in acquiring a *summary* or an *indication* of the parameters, rather than the parameters themselves. When the sensor identities are unimportant or irrelevant, it suffices to collect a summary statistic of the parameters, e.g., the arithmetic mean. Furthermore, in forest fire detection, it suffices to signal an alarm instead of collecting the whole temperature and/or humidity readings. More generally, in all these cases the fusion center is only interested in knowing a *function* of the measured parameters.

In this chapter, we consider an information-theoretic formulation of function computation over networks. The adopted performance metric is *computation rate*, i.e., the number of functions computed reliably per channel use. The adopted channel model is the Gaussian MAC, which is canonical for wireless sensor networks. In the Gaussian MAC, the sensors play the role of transmitters and the fusion center serves as the receiver. If we naively transmit all measured parameters to the fusion center, the worst-case computation rate is $\Theta\left(\frac{\log L}{L}\right)$, where L is the number of sensors.[2] However, by exploiting the superposition property of the Gaussian channel, the developed coding scheme in this chapter, termed *arithmetic computation coding*, achieves the worst-case computation rate $\Theta\left(\frac{1}{\log L}\right)$ for the following (class of) functions:

[1] Part of the material in this chapter has appeared in
S.-W. Jeon, C.-Y. Wang, and M. Gastpar, "Computation over Gaussian networks with orthogonal components," *IEEE Trans. Inf. Theory*, vol. 60, p. 7841-7861, Dec. 2014.

[2] Throughout Chapters 6 and 7, "worst-case" means the worst source distribution for computing the desired function, which may depend on L.

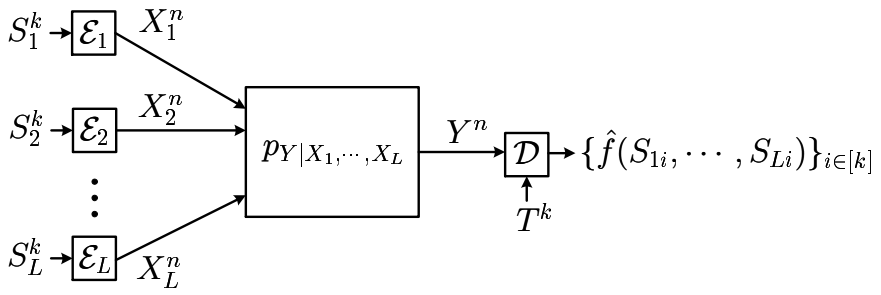

Figure 6.1: Function computation over a MAC.

- (weighted) arithmetic sum,
- (weighted) modulo-q sum,
- frequency histogram (type),
- symmetric function.

Chapter outline: First, we provide a problem statement for general MACs in Section 6.1. In Section 6.2, we consider computation of arithmetic sums over modulo adder MACs, in which the channel output is the deterministic modulo sum of the channel inputs. Then, in Sections 6.3 and 6.4, we propose efficient coding schemes for computing arithmetic sums and frequency histograms over the Gaussian MAC, respectively. Finally, we extend the arithmetic computation coding to the symmetric Rayleigh fading MAC in Section 6.5.

6.1 Problem Statement

Let L be a fixed positive integer. A DMS $\langle S_1, S_2, \cdots, S_L, T\rangle$ generates i.i.d. source sequences $(S_1^k, S_2^k, \cdots, S_L^k, T^k)$. We assume that $\mathcal{S}_\ell = \{0, \cdots, d-1\}$, where d is a positive prime integer. For convenience, we use the short-hand notation $[d]_{-1}$ to denote the set $\{0, \cdots, d-1\}$.

Now consider the multiple access communication system depicted in Figure 6.1. There are L sensors and one fusion center. At time $j \in [n]$, each sensor (indexed by $\ell \in [L]$) encodes the observed source sequence S_ℓ^k into a symbol $X_{\ell j}$ and transmits it over the shared memoryless channel governed by $p_{Y|X_1,\cdots X_L}$, a pmf for discrete alphabets or a probability density function for continuous alphabets. The fusion center has side information T^k and receives the sequence Y^n. The fusion center wishes to recover from (Y^k, T^k) an element-wise function $f(s_1, \cdots, s_L)$ losslessly.

A (k, n) block code for function computation over MAC consists of

- L encoders, where Encoder $\ell \in [L]$ assigns a symbol $x_{\ell j}(s_\ell^k) \in \mathcal{X}_\ell$ to each sequence $s_\ell^k \in [d]_{-1}^k$ for all $j \in [n]$;
- one decoder, which assigns an estimate $\hat{w}^k$ to each tuple $(y^n, t^k) \in \mathcal{Y}^n \times \mathcal{T}^k$.

We say that the computation rate $R := k/n$ is achievable if there exists a sequence of (nR, n) computation codes such that the probability of error

$$P_e^{(n)} := \mathbb{P}\left(\bigcup_{i\in[nR]} \left\{\hat{W}_i \neq f(S_{1i}, \cdots, S_{Li})\right\}\right)$$

converges to zero as n tends to infinity. Note that the computation rate is the number of reliably computed functions per channel use. Finally, the computation capacity C is the supremum over all achievable computation rates.

6.2 Compute Arithmetic Sum over Modulo Adder MACs

Let us start with a simple model to gain some insight. Consider $L = 2$. Let $S_1, S_2 \sim$ Bernoulli(1/2) and $T = \emptyset$. We assume that S_1 and S_2 are independent. We first give the definitions of $\mathbb{F}_q$-adder MAC and modulo-q adder MAC.

Definition 6.1 ($\mathbb{F}_q$-Adder MAC)**.** *Let q be a positive prime number. Then, the $\mathbb{F}_q$-adder MAC is the DMC $\langle p_{Y|X_1,X_2}\rangle$ where $|\mathcal{X}_1| = |\mathcal{X}_2| = |\mathcal{Y}| = \mathbb{F}_q$ and*

$$p_{Y|X_1,X_2}(y|x_1, x_2) = \mathbb{1}\{y = x_1 \oplus_q x_2\},$$

for all $x_1, x_2, y \in \mathbb{F}_q$.

Definition 6.2 (Modulo-q Adder MAC)**.** *Let q be a positive prime number. Then, the modulo-q adder MAC is the DMC $\langle p_{Y|X_1,X_2}\rangle$ where $|\mathcal{X}_1| = |\mathcal{X}_2| = |\mathcal{Y}| = [q]_{-1}$ and*

$$p_{Y|X_1,X_2}(y|x_1, x_2) = \mathbb{1}\{y = x_1 + x_2 \bmod q\},$$

for all $x_1, x_2, y \in [q]_{-1}$.

Note that the above definitions can be naturally extended to the general L-sensor case. Since there exists a bijection between the $\mathbb{F}_q$-adder MAC and the modulo-q adder MAC, hereafter we treat them as the same MAC. For the same reason, we consider the two sets $\mathbb{F}_q$ and $[q]_{-1}$ equivalent.

Now consider computation of the arithmetic sum $S_1 + S_2$ over the modulo-3 adder MAC. Let us first examine two basic achievable schemes:

1. Communicate the full data
   Since S_1 and S_2 are independent, communicating the full data is equivalent to transmitting two independent messages over the MAC. Then, the corresponding optimal symmetric rate is $\frac{\log(3)}{2}$. After recovering the full data, the desired arithmetic sum can be deduced and thus the computation rate $R = \frac{\log(3)}{2} \approx 0.792$ is achievable.

2. Uncoded transmission
   If we simply set $X_1 = S_1$ and $X_2 = S_2$, then the fusion center learns $Y = S_1 \oplus_3 S_2$. Since there is no "wrap around" in this case, the modulo-3 sum is equal to the arithmetic sum $S_1 + S_2$. The achievable computation rate is thus $R = 1$.

The above two schemes have their own pros and cons. Communicating the full data allows us to apply the optimal channel code, but it reveals more redundant information which leads to a low computation rate. On the other hand, uncoded transmission reduces the information redundancy, but it is a bad code from the perspective of channel coding since the symbol "2" is never used.

We now present an achievable scheme preserving the advantages of both the above coding schemes. The key ingredients of the proposed coding scheme are *linear computation coding* and *embedding*. We have the following proposition.

Proposition 6.1. *Consider computation of arithmetic sum over the modulo-*3 *adder MAC. Any computation rate* R *satisfying* $R \leq \frac{2\log(3)}{3} (\approx 1.057)$ *is achievable.*

Proof: First, Encoder $\ell \in \{1, 2\}$ embeds the source sequence $s_\ell^k \in \mathbb{F}_2^k$ into a sequence $\tilde{s}_\ell^k \in \mathbb{F}_3^k$ with the mapping $\tilde{s}_{\ell i} = s_{\ell i}$, $i \in [k]$. Then, we use linear computation coding to communicate the modulo-3 sum $\tilde{S}_1 \oplus_3 \tilde{S}_2$. Theorem 2.7 says that any computation rate R satisfying

$$R \leq \frac{I(W;Y)\Big|_{W\sim \mathrm{Uniform}(\mathbb{F}_3)}}{H\left(\tilde{S}_1 \oplus_3 \tilde{S}_2\right)} = \frac{\log(3)}{3/2}.$$

is achievable. Finally, we note that in this case $\tilde{S}_1 \oplus_3 \tilde{S}_2 = S_1 + S_2$ and thus the computation rate $R = \frac{\log(3)}{3/2}$ is achievable. ∎

We remark that if the desired function is the modulo-2 sum $S_1 \oplus S_2$ instead, then we can first compute the arithmetic sum and then perform a modulo-2 operation to recover the desired modulo-2 sum. However, it is not known whether the computation rate $R = \frac{\log(3)}{3/2}$ is optimal for computing the modulo-2 sum.

In general, by combining the linear computation coding with embedding, we have the following theorem for computing arithmetic sums over the modulo-q adder MAC. The converse follows from a simple cut-set argument.

Theorem 6.1. *Consider computation of the arithmetic sum* $\sum_{\ell=1}^L S_\ell$ *over the modulo-*q *adder MAC. If*

$$\mathbb{P}\left(\sum_{\ell=1}^L S_\ell < q\right) = 1, \tag{6.1}$$

then the computation capacity is $C = \frac{\log q}{H\left(\sum_{\ell=1}^L S_\ell\right)}$.

Note that Condition (6.1) says that "wrap around" happens almost never. Since any modulo-d sum can be derived from the arithmetic sum, we have the following corollary.

Corollary 6.1. *Consider computation of the modulo-*d *sum* $\bigoplus_{\ell=1}^L S_\ell$ *over the modulo-*q *adder MAC. If*

$$\mathbb{P}\left(\sum_{\ell=1}^L S_\ell < q\right) = 1,$$

then any computation rate R *satisfying* $R \leq \frac{\log q}{H\left(\sum_{\ell=1}^L S_\ell\right)}$ *is achievable.*

6.3 Arithmetic Computation Coding over the Gaussian MAC

In this and the next section, we consider the Gaussian MAC which has the following input–output relation:

$$Y = \sum_{\ell=1}^{L} h_\ell X_\ell + Z,$$

where $h_\ell \in \mathbb{R}$, $\ell \in [L]$, are fixed constants, $Z \sim \mathcal{N}(0,1)$, and $X_\ell \in \mathbb{R}$ for all $\ell \in [L]$. We assume that the channel coefficient vector $\mathbf{h}$ is known at all nodes. Additionally, each encoder (indexed by $\ell \in [L]$) needs to satisfy the average power constraint

$$\frac{1}{n}\sum_{j=1}^{n} x_{\ell j}^2 \le P,$$

for some fixed $P > 0$.

In the previous section, we have seen that linear computation coding with embedding performs well in computing arithmetic sums over modulo-q adder MACs as long as there is no "wrap around." To extend to the Gaussian MAC, we need a scheme to bridge the modulo adder MACs and the Gaussian MAC. One such bridge is *nested lattice codes*, which is used to develop the compute-and-forward framework [50]. Instead of delving into the details of nested lattice codes, we briefly summarize how to interpret the results offered by the compute-and-forward framework so that we may use them for computing arithmetic sums over the Gaussian MAC.

Considering the binary presentation of $f(S_1, \cdots, S_L)$, it is meaningful to define *computation bit rate* $R_{\mathsf{bit}} := H(f(S_1, \cdots, S_L))R$, which is the number of computed bits per channel use.[3] Rather than computing a fixed function, the compute-and-forward targets on maximizing the computation bit rate R_{bit} over a class of functions for independent and uniformly distributed messages $(M_1, \cdots, M_L) \in [2^{nR_{\mathsf{bit}}}]^L$.

Fix $\mathbf{a} \in \mathbb{Z}^L$. To each prime number q, we attribute a function $f_q(M_1, \cdots, M_L) = Y'_{[t]}$, where $t = \frac{nR_{\mathsf{bit}}}{\log q}$ and

$$Y'_j = \bigoplus_{\ell=1}^{L} (a_\ell \bmod q) \otimes X'_{\ell j},$$

in which all operations are over $\mathbb{F}_q$, and $X'_{\ell j}$ is the j-th entry of the q-ary representation of M_ℓ. Then, Theorem 2 in [50] can be restated as follows.

Theorem 6.2 (Nazer and Gastpar)**.** *Let $\mathbf{a} \in \mathbb{Z}^L$ be fixed. If it suffices to compute reliably any one of the functions $\{f_q(M_1, \cdots, M_L)\}_{\{q \text{ is prime}\}}$ over the Gaussian MAC described in* (6.2)*, then any computation bit rate R_{bit} satisfying*

$$R_{\mathsf{bit}} < \frac{1}{2}\log^+\left(\frac{1 + P\|\mathbf{h}\|^2}{\|\mathbf{a}\|^2 + P(\|\mathbf{h}\|^2\|\mathbf{a}\|^2 - |\mathbf{h}^T\mathbf{a}|^2)}\right) \tag{6.2}$$

is achievable.

[3] The defined computation bit rate is equivalent to the computation rate considered in [50].

Since the coefficient vector $\mathbf{h}$ is available at all nodes, the sensors can apply *beamforming* to achieve a higher computation bit rate and we have the following corollary.

Corollary 6.2. *Let $\mathbf{a} \in \mathbb{Z}^L$ be fixed. If it suffices to compute reliably any one of the functions $\{f_q(M_1, \cdots, M_L)\}_{\{q \text{ is prime}\}}$ over the Gaussian MAC described in* (6.2)*, then any computation bit rate R_{bit} smaller than*

$$R^*_{\mathsf{bit}}(\mathbf{h}, \mathbf{a}) := \sup_{\mathbf{u}} \frac{1}{2} \log^+ \left(\frac{1 + P\|\mathbf{u} \circ \mathbf{h}\|^2}{\|\mathbf{a}\|^2 + P(\|\mathbf{u} \circ \mathbf{h}\|^2 \|\mathbf{a}\|^2 - |(\mathbf{u} \circ \mathbf{h})^T \mathbf{a}|^2)} \right) \tag{6.3}$$

is achievable, where $\mathbf{u} \circ \mathbf{h} = (u_1 h_1, \cdots, u_L h_L)^T$ and the supremum is over all $\mathbf{u} \in [-1, 1]^L$ such that for all $\ell \in [L]$, $u_\ell = 0$ if and only if $a_\ell = 0$.

Remark 6.1. *It suffices to assume that $\mathbf{a}$ is set-wise coprime, i.e., there exists no integer $c > 1$ such that $a_\ell \bmod c = 0$ for all $\ell \in [L]$. The reason is that if such c exists, we can first pretend that the coefficient vector is $\mathbf{a}/c$, and then multiply the recovered modulo sum with c at the fusion center. Besides, as can be seen from* (6.3)*, we can acquire an additional gain $\frac{1}{2} \log c$ in computation bit rate. For the ease of exposition, we implicitly assume that $\mathbf{a}$ is set-wise coprime hereafter.*

On the other hand, by setting $u_\ell = \frac{\min_{i \in [L]} |h_i/a_i|}{h_\ell / a_\ell}$, $\ell \in [L]$, we have

$$R^*_{\mathsf{bit}}(\mathbf{h}, \mathbf{a}) \geq \frac{1}{2} \log^+ \left(\frac{1}{\|\mathbf{a}\|^2} + \left(\min_{\ell \in [L]} \left| \frac{h_\ell}{a_\ell} \right| \right) P \right), \tag{6.4}$$

which is tight in the high power regime.

To achieve a computation bit rate arbitrarily close to $R^*_{\mathsf{bit}}(\mathbf{h}, \mathbf{a})$ for all P, currently the only way is to send q to infinity. This fact is not in favor of computing modulo sums, but it is useful for computing arithmetic sums as the following theorem shows.

Theorem 6.3. *Fix $\mathbf{c} \in \mathbb{N}^L$. Consider computation of the function*

$$f(s_1, \cdots, s_L) = \sum_{\ell=1}^{L} c_\ell s_\ell$$

over the Gaussian MAC described in (6.2)*. Any computation rate R satisfying*

$$R < \frac{R^*_{\mathsf{bit}}(\mathbf{h}, \mathbf{a})}{H\left(\sum_{\ell=1}^{L} c_\ell S_\ell \middle| T \right)}$$

is achievable, where $a_\ell = \mathbb{1}\{c_\ell \neq 0\}$, $\ell \in [L]$.

Proof: Assume that $q > d \sum_{\ell=1}^{L} |c_\ell|$. For convenience, we denote $W = \bigoplus_{\ell=1}^{L} (c_\ell S_\ell \bmod q)$.

Codebook: Let the matrix $\mathbf{H}$ be of size $kH(W|T) \times k$ and let $\mathbf{d}_\ell$ be a length-kR vector, $\ell \in [L]$. All entries are i.i.d. drawn from Uniform($\mathbb{F}_q$). Besides, we prepare a lattice codebook resulting from a good nested lattice code. We assume that $\mathbf{H}$, $\mathbf{d}_{[L]}$, and the lattice codebook are revealed to all nodes.

Encoding: Upon observing the source sequence $s_\ell^k \in [d]_{-1}^k$, Encoder $\ell \in [L]$ remains silent if $c_\ell = 0$. Otherwise, Encoder ℓ embeds s_ℓ^k into a sequence $\tilde{s}_\ell^k \in \mathbb{F}_q^k$ with the mapping $\tilde{s}_{\ell i} = c_\ell s_{\ell i}$, $i \in [k]$. Then, Encoder $\ell \in [L]$ performs linear binning and dithering:

$$m_\ell = \mathbf{H}\tilde{\mathbf{s}}_\ell \oplus_q \mathbf{d}_\ell.$$

Then, Encoder $\ell \in [L]$ maps m_ℓ to a codeword $x_\ell^n(m_\ell)$ using the nested lattice code and transmits $x_\ell^n(m_\ell)$.

Decoding: Upon receiving y^n, the decoder first applies lattice decoding to recover the weighted modulo-q sum vector

$$\begin{aligned} m_{\mathsf{sum}} &= \bigoplus_{\ell:c_\ell \neq 0} m_\ell \\ &= \bigoplus_{\ell:c_\ell \neq 0} (\mathbf{H}\tilde{\mathbf{s}}_\ell \oplus_q \mathbf{d}_\ell) \\ &= \mathbf{H}\left(\bigoplus_{\ell:c_\ell \neq 0} \tilde{\mathbf{s}}_\ell\right) \oplus_q \bigoplus_{\ell:c_\ell \neq 0} \mathbf{d}_\ell. \end{aligned}$$

After removing the dithers, the decoder applies joint typicality decoding with side information t^k to recover $\hat{w}^k$.

Using the distributive property of modulo operation, we have that for all $i \in [k]$,

$$\begin{aligned} \bigoplus_{\ell:c_\ell \neq 0} \tilde{s}_{\ell i} &= \sum_{\ell=1}^{L} c_\ell s_{\ell i} \bmod q \\ &\overset{(a)}{=} \sum_{\ell=1}^{L} c_\ell s_{\ell i}, \end{aligned}$$

where (a) follows since we assume $q > d\sum_{\ell=1}^{L} |c_\ell|$. Finally, Theorems 2.7 and 6.2 imply that as q tends to infinity, any computation rate R satisfying

$$R < \frac{R_{\mathsf{bit}}^*(\mathbf{h}, \mathbf{a})}{H\left(\sum_{\ell=1}^{L} c_\ell S_\ell \middle| T\right)}$$

is achievable. ■

Next, we use the facts that any modulo-d sum can be deduced from the arithmetic sum and that $\mathbb{F}_d$ is closed under division to develop a coding scheme for computing modulo-d sums over the Gaussian MAC.

Theorem 6.4. *Fix $\mathbf{c} \in \mathbb{F}_d^L$. Consider computation of the function*

$$f(s_1, \cdots, s_L) = \bigoplus_{\ell=1}^{L} c_\ell \otimes_d s_\ell$$

over the Gaussian MAC described in (6.2)*. Any computation rate R satisfying*

$$R < \max_{\mathbf{a}} \frac{R_{\mathsf{bit}}^*(\mathbf{h}, \mathbf{a})}{H\left(\sum_{\ell=1}^{L} a_\ell (c_\ell \otimes_d \beta_\ell \otimes_d S_\ell) \middle| T\right)}$$

is achievable, where $\beta_\ell \in \mathbb{F}_d$ is the inverse of $a_\ell \bmod d$ and the maximum is over all $\mathbf{a} \in \mathbb{Z}^L$ satisfying $a_\ell = 0$ if and only if $c_\ell = 0$ for all $\ell \in [L]$.

Proof: Consider any $\mathbf{a} \in \mathbb{Z}^L$ satisfying $a_\ell = 0$ if and only if $c_\ell = 0$ for all $\ell \in [L]$. Assume that $q \geq 2d\sum_{\ell=1}^{L}|a_\ell| + 1$. For convenience, we denote $W = \bigoplus_{\ell=1}^{L}(a_\ell \bmod q)(c_\ell \otimes_d \beta_\ell \otimes_d S_\ell)$.

Codebook: Let the matrix $\mathbf{H}$ be of size $kH(W|T) \times k$ and let $\mathbf{d}_\ell$ be a length-kR vector, $\ell \in [L]$. All entries are i.i.d. drawn from Uniform($\mathbb{F}_q$). Besides, we prepare a lattice codebook resulting from a good nested lattice code. We assume that $\mathbf{H}$, $\mathbf{d}_{[L]}$, and the lattice codebook are revealed to all nodes.

Encoding: Upon observing the source sequence $s_\ell^k \in [d]_{-1}^k$, Encoder $\ell \in [L]$ embeds s_ℓ^k into a sequence $\tilde{s}_\ell^k \in \mathbb{F}_q^k$ with the mapping $\tilde{s}_{\ell i} = c_\ell \otimes_d \beta_\ell \otimes_d s_\ell$, $i \in [k]$. Then, Encoder $\ell \in [L]$ performs linear binning and dithering:

$$m_\ell = \mathbf{H}\tilde{\mathbf{s}}_\ell \oplus_q \mathbf{d}_\ell.$$

Then, Encoder $\ell \in [L]$ maps m_ℓ to a codeword $x_\ell^n(m_\ell)$ using the nested lattice code and transmits $x_\ell^n(m_\ell)$.

Decoding: Upon receiving y^n, the decoder first applies lattice decoding to recover the weighted modulo-q sum vector

$$\begin{aligned} m_{\mathsf{sum}} &= \bigoplus_{\ell=1}^{L}(a_\ell \bmod q) \otimes_q m_\ell \\ &= \bigoplus_{\ell=1}^{L}(a_\ell \bmod q) \otimes_q (\mathbf{H}\tilde{\mathbf{s}}_\ell \oplus_q \mathbf{d}_\ell) \\ &= \mathbf{H}\left(\bigoplus_{\ell=1}^{L}(a_\ell \bmod q) \otimes_q \tilde{\mathbf{s}}_\ell\right) \oplus_q \bigoplus_{\ell=1}^{L}(a_\ell \bmod q) \otimes_q \mathbf{d}_\ell. \end{aligned}$$

After removing the dithers, the decoder applies joint typicality decoding with side information t^k to recover $\hat{w}^k$.

Using the distributive property of modulo operation, we have for all $i \in [k]$,

$$\begin{aligned} \bigoplus_{\ell=1}^{L}(a_\ell \bmod q) \otimes_q \tilde{s}_{\ell i} &= \sum_{\ell=1}^{L}(a_\ell \bmod q) \otimes_q (c_\ell \otimes_d \beta_\ell \otimes_d s_{\ell i}) \\ &= \sum_{\ell=1}^{L} a_\ell(c_\ell \otimes_d \beta_\ell \otimes_d s_{\ell i}) \mod q. \end{aligned}$$

Since we assume $q \geq 2d\sum_{\ell=1}^{L}|a_\ell|+1$, there is a bijection between the arithmetic sum $\sum_{\ell=1}^{L} a_\ell(c_\ell \otimes_d \beta_\ell \otimes_d s_{\ell i})$ and the corresponding modulo-q sum. Thus, the arithmetic sum can be recovered from the modulo-q sum. Then, taking modulo-d operation, we have

$$\begin{aligned} \left(\sum_{\ell=1}^{L} a_\ell(c_\ell \otimes_d \beta_\ell \otimes_d s_{\ell i})\right) \bmod d &= \bigoplus_{\ell=1}^{L}(a_\ell \bmod d) \otimes_d (c_\ell \otimes_d \beta_\ell \otimes_d s_{\ell i}) \\ &\overset{(a)}{=} \bigoplus_{\ell=1}^{L} c_\ell \otimes_d s_{\ell i}, \end{aligned}$$

where (a) follows since β_ℓ is the inverse of $a_\ell \bmod d$. Finally, Theorems 2.7 and 6.2 imply that as q tends to infinity, any computation rate R satisfying

$$R < \frac{R_{\mathsf{bit}}^*(\mathbf{h}, \mathbf{a})}{H\left(\sum_{\ell=1}^{L} a_\ell(c_\ell \otimes_d \beta_\ell \otimes_d S_\ell)\middle| T\right)}$$

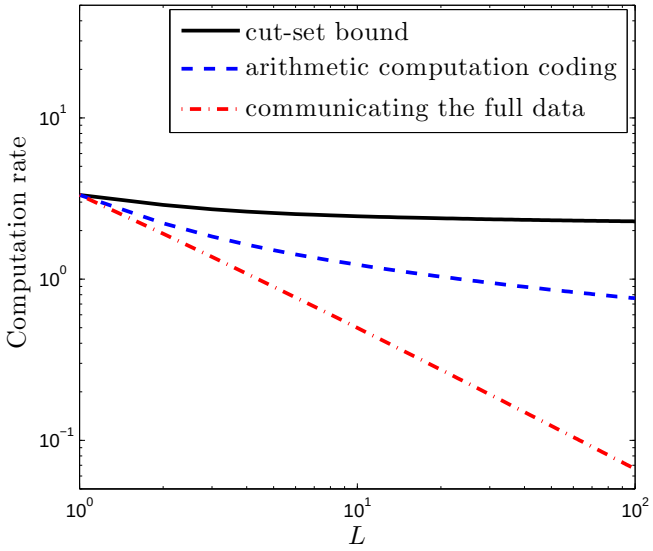

Figure 6.2: Computation of the arithmetic mean $f(s_1, \cdots, s_L) = \frac{1}{L}\sum_{\ell=1}^{L} s_\ell$ over the Gaussian MAC with equal channel gains. The power constraint is $P = 20$ dB.

is achievable, and the theorem follows by optimizing (6.5) over all $\mathbf{a} \in \mathbb{Z}^L$ satisfying $a_\ell = 0$ if and only if $c_\ell = 0$ for all $\ell \in [L]$. ■

We refer to the coding scheme used in proving Theorems 6.3 and 6.4 as *arithmetic computation coding*. Let us present an example that demonstrates the advantage of arithmetic computation coding over the simple "communicating the full data" scheme.

Example 6.1. *Assume that $(S_1, \cdots, S_L)$ are i.i.d. drawn from Bernoulli(1/2), $T = \emptyset$, and $h_\ell = 1$ for all $\ell \in [L]$. The fusion center wishes to recover the arithmetic mean $f(s_1, \cdots, s_L) = \frac{1}{L}\sum_{\ell=1}^{L} s_\ell$. Let $W = \sum_{\ell=1}^{L} S_\ell$. Note that $W \sim$ Binomial$(L, 1/2)$. Then, the arithmetic computation coding achieves any computation rate R satisfying*

$$R < \frac{\frac{1}{2}\log^+\left(\frac{1}{L} + P\right)}{H(W)}.$$

On the other hand, the cut-set bound and the "communicating the full data" scheme together give the the following bound on the computation capacity:

$$\frac{\frac{1}{2}\log(1+LP)}{L} < C \le \frac{\frac{1}{2}\log\left(1+L^2P\right)}{H(W)}.$$

Figure 6.2 plots the three bounds with respect to the number of sensors L, where $P = 20$ dB. Since $H(W)$ scales as $\Theta(\log L)$ as L increases, the arithmetic computation coding, the "communicating the full data" scheme, and the cut-set bound have the scaling $\Theta\left(\frac{\log P}{\log L}\right)$, $\Theta\left(\frac{\log LP}{L}\right)$, and $\Theta\left(\frac{\log LP}{\log L}\right)$, respectively.

6.4 Compute Frequency Histogram (Type) over the Gaussian MAC

In this section, we show that by expanding the framework of arithmetic computation coding, we can compute frequency histograms and all symmetric functions more efficiently. We exploit the fact that the sensors and the fusion center can perform some computations locally and then we have the following theorem.

Theorem 6.5. *Assume that a function $f : [d]_{-1}^{L} \to \Lambda$ can be expressed as*

$$f(s_1, \cdots, s_L) = \phi\left(\sum_{\ell=1}^{L} c_\ell \psi_\ell(s_\ell)\right), \tag{6.5}$$

for some $\mathbf{c} \in \mathbb{N}^L$ and some functions $\phi : \mathbb{Z} \to \Lambda$ and $\psi_\ell : [d]_{-1} \to \mathbb{Z}$, $\ell \in [L]$. Consider computation of the function f over the Gaussian MAC. Then, any computation rate R satisfying

$$R < \frac{R_{\mathsf{bit}}^{*}(\mathbf{h}, \mathbf{a})}{H\left(\sum_{\ell=1}^{L} c_\ell \psi_\ell(S_\ell) \middle| T\right)}$$

is achievable, where $a_\ell = \mathbb{1}\{c_\ell \neq 0\}$, $\ell \in [L]$.

We have two comments on Expression (6.5). First, the representation is not unique. Second, every function $f : [d]_{-1}^{L} \to \Lambda$ can be expressed as (6.5) since one can sophisticatedly choose $\mathbf{c}$ and $\psi_{[L]}$ such that $(s_1, \cdots, s_L)$ can be deduced from the arithmetic sum $\sum_{\ell=1}^{L} c_\ell \psi(s_\ell)$. Clearly, this is a poor choice since too much redundant information will be revealed. Therefore, in order to maximize the achievable computation rate, we need to carefully choose $\mathbf{c}$ and $\psi_{[L]}$. In the ideal situation, there exist $\mathbf{c}$ and $\psi_{[L]}$ such that the resulting ϕ can be a bijection.

Definition 6.3 (Frequency Histogram, Type)**.** *The frequency histogram (or type) of a sequence $s_{[L]} \in [d]_{-1}^{L}$ is a length-d vector $b_{[d]_{-1}}$ with*

$$b_j := \sum_{\ell=1}^{L} \mathbb{1}\{s_\ell = j\}.$$

The b_j is termed frequency of j.

Definition 6.4 (Symmetric Function)**.** *Let Λ be a finite alphabet. A function $f : [d]_{-1}^{L} \to \Lambda$ is called symmetric if*

$$f(s_{\sigma(1)}, s_{\sigma(2)}, \cdots, s_{\sigma(L)}) = f(s_1, s_2, \cdots, s_L),$$

for every permutation σ on $[L]$.

Note that every symmetric function can be deduced from the frequency histogram.

Now we are ready to demonstrate how to communicate frequency histograms (types) and any symmetric function efficiently over the Gaussian MAC. A natural representation of b_j in the form of (6.5) is to set $c_\ell = 1$, $\psi_\ell(s_\ell) = \mathbb{1}\{s_\ell = j\}$ for all $\ell \in [L]$, and $\phi(x) = x$, i.e., the identity function. Since the identity function is clearly a

bijection, a frequency can be efficiently communicated using arithmetic computation coding. Furthermore, the entire frequency histogram can be communicated to the fusion center by conveying frequencies one by one. The recovered frequencies in the previous rounds can be treated as side information for the current round. Once the frequency histogram is recovered, we can compute any desired symmetric function. Thus, we have the following corollary.

Corollary 6.3. *Consider computation of the frequency histogram or any symmetric function over the Gaussian MAC. Any computation rate R satisfying*

$$R < \frac{R^*_{\mathsf{bit}}(\mathbf{h}, \mathbf{1})}{H\left(B_0, \cdots, B_{d-1}\right)}$$

is achievable.

Finally, we remark that the worst-case scaling of the entropy of frequency histogram is $\Theta(\log L)$. Thus, comparing with the worst-case scaling of the entropy of full data $\Theta(L)$, the arithmetic computation coding greatly reduces the amount of redundant information.

6.5 Computation over the Symmetric Rayleigh Fading MAC

In this section, we consider the symmetric Rayleigh fading MAC, which has the following input–output relation:

$$Y = \sum_{\ell=1}^{L} H_\ell X_\ell + Z, \tag{6.6}$$

where $X_\ell \in \mathbb{C}$ for all $\ell \in [L]$, the complex channel coefficients $H_{[L]}$ are i.i.d. drawn from the circularly-symmetric complex Gaussian distribution $\mathcal{CN}(0,1)$ and $Z \sim \mathcal{CN}(0,1)$ is independent of $H_{[L]}$. We assume that the channel coefficients $H_{[L]}$ are known at all nodes. Additionally, each encoder (indexed by $\ell \in [L]$) needs to satisfy the average power constraint

$$\frac{1}{n}\sum_{j=1}^{n} |x_{\ell j}|^2 \le P,$$

for some fixed $P > 0$.

The development for the complex-valued channel model is similar to the real-valued counterpart, so the details are omitted. One main difference at the encoding side is that the message generated from linear computation coding with embedding is divided into two equal parts and passed to the complex-valued compute-and-forward (see [50, Theorems 3 and 4]).

In the high power regime, it is desired that $\mathbf{a}$ and $\mathbf{h}$ are aligned, i.e., $\mathbf{a} = c\mathbf{h}$ for some c, so that the achievable computation rate scales logarithmically with P. Although we can apply beamforming to force $\mathbf{u} \circ \mathbf{h}$ to align with $\mathbf{a}$, as done in Remark 6.1, most of power ends up unused at most sensors, depending on the amplitudes of $u_{[L]}$.

In the fast fading scenario, we can perform adaptive power allocation to fully utilize the available power at the sensor nodes. That is, the amplitudes of the entries of the beamforming vector is not restricted to one. We modify Theorem 6.3 to the fading scenario in a similar approach as [51]. Here we consider the achievable *ergodic* computation bit rate. Then, the ergodic computation rate can be similarly derived as in Theorem 6.5.

We assume $\mathbf{a} = \mathbf{1}$. At each time slot, Sensor $\ell \in [L]$ sets

$$U_\ell = \frac{\overline{H}_\ell \min_{i\in[L]} |H_i|/|H_\ell|^2}{\sqrt{\mathbb{E}[\min_{i\in[L]} |H_i|^2/|H_\ell|^2]}}, \tag{6.7}$$

where $\overline{H}_\ell$ is the complex conjugate of H_ℓ. Then, combining with Expression (6.4), any ergodic computation bit rate R_{bit} satisfying

$$R_{\mathsf{bit}} < \mathbb{E}\left[\log^+\left(\frac{1}{L} + \frac{\min_{i\in[L]} |H_i|^2}{\mathbb{E}[\min_{i\in[L]} |H_i|^2/|H_1|^2]} P\right)\right] =: R_{\mathsf{lower}}$$

is achievable.

On the other hand, the cut-set bound is given by

$$R_{\mathsf{bit}} \le \max_{\phi_1,\cdots,\phi_L} \mathbb{E}\left[\log\left(1 + \left(\sum_{\ell=1}^{L} |H_\ell|\phi_\ell(H_1,\cdots,H_L)\right)^2\right)\right] =: R_{\mathsf{upper}},$$

where ϕ_ℓ is the power allocation policy adopted by Sensor ℓ and satisfies that $\mathbb{E}[\phi_\ell^2(H_1,\cdots,H_L)] \le P$. Denote by ϕ_ℓ^* the optimal power allocation policy at Sensor $\ell \in [L]$.

The following proposition shows that with the choice of beamforming vector in (6.8) and $\mathbf{a} = \mathbf{1}$, the achievable ergodic computation bit rate in (6.7) has a constant gap, independent of P, from the optimal computation bit rate.

Proposition 6.2. $R_{\mathsf{upper}} - R_{\mathsf{lower}} \le 2\log L + \log(\ln L) + 3 + \log e$.

Proof: Before bounding the difference between R_{upper} and R_{lower}, we evaluate $\mathbb{E}\left[\min_{i\in[L]} |H_i|^2/|H_1|^2\right]$ and find a lower bound on $\mathbb{E}\left[\log\left(\min_{i\in[L]} |H_i|^2\right)\right]$. For convenience, denote $U = |H_1|^2$ and $V = \min_{i\in[L]\setminus\{1\}} |H_\ell|^2$. Since $U \sim$ Exponential(1) and $V \sim$ Exponential$(L-1)$, we have

$$\begin{aligned}
& \mathbb{E}\left[\min_{i\in[L]} |H_i|^2/|H_1|^2\right] \\
= \;& \mathbb{E}[\min\{U,V\}/U] \\
= \;& \int_0^\infty \int_0^\infty \frac{\min\{u,v\}}{u} e^{-u}(L-1)e^{-(L-1)v}\, du dv \\
= \;& \int_0^\infty e^{-u}\left[\int_0^u \frac{v}{u}(L-1)e^{-(L-1)v}\, dv + \int_u^\infty (L-1)e^{-(L-1)v}\, dv\right] du \\
= \;& \int_0^\infty e^{-u}\left[\left(\frac{1}{(L-1)u} - \left(1 + \frac{1}{(L-1)u}\right)e^{-(L-1)u}\right) + e^{-(L-1)u}\right] du \\
= \;& \frac{1}{L-1}\int_0^\infty \frac{e^{-u} - e^{-Lu}}{u}\, du
\end{aligned}$$

$$\begin{aligned}
&\overset{\nu=e^{-u}}{=} \frac{1}{L-1}\int_0^1 \frac{\nu^{L-1}-1}{\ln \nu}\,d\nu \\
&\overset{(a)}{=} \frac{\ln L}{L-1},
\end{aligned} \tag{6.8}$$

where (a) follows from [52, 4.267 8]. Next, we have

$$\begin{aligned}
\mathbb{E}\left[\log\left(\min_{i\in[L]}|H_i|^2\right)\right] &= \int_0^\infty \log(u) L e^{-Lu}\,du \\
&\overset{\nu=Lu}{=} -\log L + \int_0^\infty \log(\nu) e^{-\nu}\,d\nu \\
&\geq -\log L + \int_0^1 \log(\nu)\,d\nu + \int_1^\infty \log(\nu) e^{-\nu}\,d\nu \\
&\geq -\log L - \log e.
\end{aligned}$$

Now we can bound the difference between R_{upper} and R_{lower} as follows:

$$\begin{aligned}
&R_{\mathsf{upper}} - R_{\mathsf{lower}} \\
&\overset{(a)}{\leq} \mathbb{E}\left[\log\left(\frac{1+\left(\sum_{\ell=1}^L |H_\ell|\phi_\ell(H_1,\cdots,H_L)\right)^2}{1+\frac{L(L-1)\min_{i\in[L]}|H_i|^2}{\ln L}P}\right)\right] + \log L \\
&\leq \mathbb{E}\left[\log\left(\frac{\frac{L(L-1)\min_{i\in[L]}|H_i|^2}{\ln L} + \frac{1}{P}\left(\sum_{\ell=1}^L |H_\ell|\phi_\ell(H_1,\cdots,H_L)\right)^2}{\frac{L(L-1)\min_{i\in[L]}|H_i|^2}{\ln L}}\right)\right] + \log L \\
&\overset{(b)}{\leq} \mathbb{E}\left[\log\left(\frac{L(L-1)\min_{i\in[L]}|H_i|^2}{\ln L} + \frac{1}{P}\left(\sum_{\ell=1}^L |H_\ell|\phi_\ell(H_1,\cdots,H_L)\right)^2\right)\right] \\
&\quad +(\log L + \log e) - \log\left(\frac{L-1}{\ln L}\right) \\
&\overset{(c)}{\leq} 2\mathbb{E}\left[\log\left(\sqrt{\frac{L(L-1)\min_{i\in[L]}|H_i|^2}{\ln L}} + \frac{1}{\sqrt{P}}\sum_{\ell=1}^L |H_\ell|\phi_\ell(H_1,\cdots,H_L)\right)\right] \\
&\quad + \log\left(\frac{eL\ln L}{L-1}\right) \\
&\overset{(d)}{\leq} 2\log\left(\sqrt{\frac{L(L-1)\mathbb{E}\left[\min_{i\in[L]}|H_i|^2\right]}{\ln L}} + \frac{1}{\sqrt{P}}\sum_{\ell=1}^L \mathbb{E}\left[|H_\ell|\phi_\ell(H_1,\cdots,H_L)\right]\right) \\
&\quad + \log\left(\frac{eL\ln L}{L-1}\right) \\
&\overset{(e)}{\leq} 2\log\left(\sqrt{\frac{L-1}{\ln L}} + L\right) + \log\left(\frac{eL\ln L}{L-1}\right) \\
&\leq 2\log L + \log(\ln L) + 3 + \log e,
\end{aligned}$$

where (a) follows from (6.8) and the fact that $\log^+(u) \geq \log(u)$, (b) follows from (6.9), (c) follows since $\log(u+v) \leq 2\log(\sqrt{u}+\sqrt{v})$ for $u \geq 0$ and $v \geq 0$, (d) follows from Jensen's inequality, and (e) follows since $\mathbb{E}\left[\min_{i\in[L]}|H_i|^2\right]) = 1/L$ and

$$\mathbb{E}\left[|H_\ell|\phi_\ell^*(H_1,\cdots,H_L)\right] \leq \sqrt{\mathbb{E}[|H_\ell|^2]\mathbb{E}\left[(\phi_\ell^*(H_1,\cdots,H_L))^2\right]} \leq \sqrt{P}.$$

■

Computation over the Gaussian MAC with Feedback

7

Wireless networks follow the principle of channel reciprocity, i.e., the channel coefficient from Antenna 1 to Antenna 2 is the same as the channel coefficient from Antenna 2 to Antenna 1.[1] Therefore, the fusion center not only receives the data sent by the sensors, but the fusion center is also capable of broadcasting messages in order to coordinate the sensors. Therefore, we can think of the fusion center as a bridge which enables interaction among the sensors. A natural question to ask is whether interaction helps in increasing the computation rate.

It turns out that interaction is beneficial for type-threshold functions, which is a subclass of symmetric functions and was first introduced in [53]. The class of type-threshold functions include the maximum, minimum, and indicator functions as special cases. Intuitively, type-threshold functions have *relatively small* ranges. For example, no matter how many dice are thrown, the maximum among them lies in the set $\{1, 2, 3, 4, 5, 6\}$.

As a proof of point, we consider the model of Gaussian MAC with noiseless causal feedback, in which the feedback link is a simplified model for the wireless channel from the fusion center to the sensors. Furthermore, we assume that the sources are independent and that the channel gains are equal, i.e., $\mathbf{h} = \mathbf{1}$. In this case, it can be shown that (the expression of $R^*_{\mathsf{bit}}(\mathbf{1}, \mathbf{a})$ is given in (6.3))

$$R^*_{\mathsf{bit}}(\mathbf{1}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{1}{\|\mathbf{a}\|^2} + P \right), \tag{7.1}$$

for all $\mathbf{a} \in \{0, 1\}^L$.

In distributed function compression of independent sources, some functions require the sources to be received in their entirety [4, Lemma 1], [6]. Amongst them, one can find many examples of type-threshold functions. This insight implies results on the worst-case scaling for computing type-threshold functions over various communication models. First, for the collision MAC, where concurrent transmissions by multiple nodes result in collisions and only the destination receives any signals,

[1] The material of this chapter has appeared in
C.-Y. Wang, S.-W. Jeon, and M. Gastpar, "Interactive computation of type-threshold functions in collocated Gaussian networks," *IEEE Trans. Inf. Theory*, vol. 61, p. 4765-4775, Sep. 2015.

Table 7.1: Worst-case scaling laws for the number of sensors.

	Full data	Symmetric functions	Type-threshold functions
Collision MAC	$\Theta\left(\frac{1}{L}\right)$	$\Theta\left(\frac{1}{L}\right)$ [4]	$\Theta\left(\frac{1}{L}\right)$ [4]
Collision MAC with Feedback	$\Theta\left(\frac{1}{L}\right)$	$\Theta\left(\frac{1}{L}\right)$ [53]	$\Theta\left(\frac{1}{\log L}\right)$ [53]
Gaussian MAC	$\Theta\left(\frac{\log LP}{L}\right)$	$\Omega\left(\frac{\log P}{\log L}\right)$ (Corollary 6.3)	$\Omega\left(\frac{\log P}{\log L}\right)$ (Corollary 6.3)
Gaussian MAC with Feedback	$\Theta\left(\frac{\log LP}{L}\right)$	$\Omega\left(\frac{\log P}{\log L}\right)$ (Corollary 6.3)	$\Omega\left(\log P\right)$ (Corollary 7.1)

the worst-case scaling is $\Theta\left(\frac{1}{L}\right)$. Next, if we consider a collision MAC with noiseless causal feedback, where feedback enables *interaction* among the encoding terminals, the worst-case scaling is $\Theta\left(\frac{1}{\log L}\right)$, see [53, 54]. On the other hand, it is interesting to consider communication models capturing the *superposition* property of wireless networks. One canonical example is the Gaussian MAC considered in Chapter 6. We showed that for all symmetric functions and thus for all type-threshold functions, the worst-case scaling is at least $\Omega\left(\frac{1}{\log L}\right)$. Apparently, in all these cases, the worst-case computation rate vanishes as the number of terminals increases.

The main result in this chapter shows that equipped with *both* interaction and superposition, the worst-case computation rate no longer vanishes as the number of terminals increases. We propose a novel coding scheme termed *multi-round group broadcast*, which is an extension of arithmetic computation coding developed in Chapter 6 to the framework of interactive computation. We show that, for any independent source distribution, all type-threshold functions are reliably computable with a non-vanishing rate over the Gaussian MAC with noiseless causal feedback, even if the number of sensors tends to infinity. That is, the worst-case scaling is at least $\Omega(1)$. Table 7.1 summarizes the worst-case scaling laws for various functions under different network models. For the Gaussian models, the average power constraint P is also presented for reference. For both the Gaussian MAC and the Gaussian MAC with feedback, a simple cut-set argument gives $O\left(\frac{\log LP}{\log L}\right)$ for symmetric functions and $O(\log LP)$ for type-threshold functions in the worst case.[2]

Chapter Outline: In Section 7.1, we provide the problem formulation defining the network model and type-threshold functions. In Section 7.2, we introduce a set of auxiliary random variables, also termed *descriptions*, with an analysis on its entropy. These descriptions serve as the building blocks for interactive coding for computing. Building upon the arithmetic computation coding in Chapter 6 and the introduced descriptions, the proposed multi-round group broadcast is presented in Section 7.3. For completeness, we provide a simple cut-set based upper bound in Section 7.4.

[2] In fact, for computing type-threshold functions over the Gaussian MAC, the upper bound on the worst-case scaling can be tightened to $O(\log P)$.

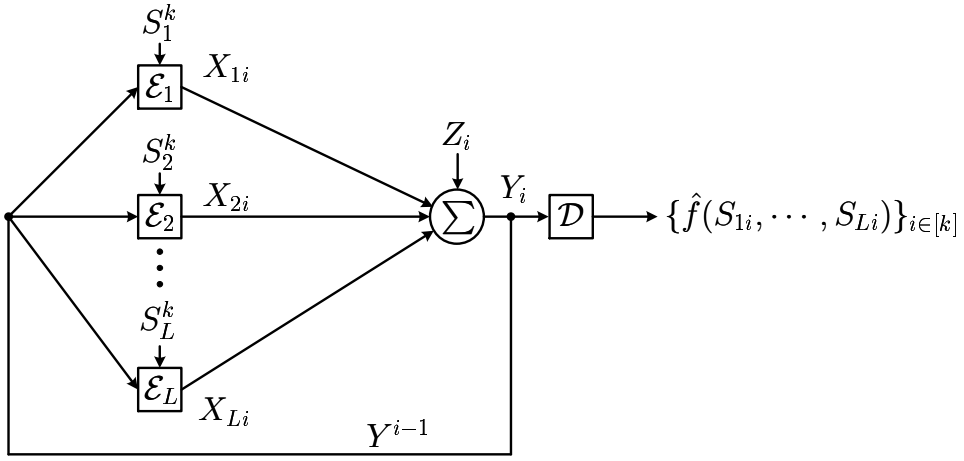

Figure 7.1: Function computation over the Gaussian MAC with noiseless causal feedback.

7.1 Problem Statement

Let L be a fixed positive integer. A DMS $\langle S_1, S_2, \cdots, S_L\rangle$ generates i.i.d. source sequences $(S_1^k, S_2^k, \cdots, S_L^k)$. We assume that $\mathcal{S}_\ell = \{0, \cdots, d-1\}$, where d is a positive prime integer. For convenience, we use the short-hand notation $[d]_{-1}$ to denote the set $\{0, \cdots, d-1\}$. In this chapter we assume that $p_{S_1,\cdots,S_L} = \prod_{\ell=1}^{L} p_{S_\ell}$, i.e., the sources are independent.

Now consider the Gaussian MAC with noiseless causal feedback depicted in Figure 7.1. There are L sensors and one fusion center. At time $j \in [n]$, each sensor (indexed by $\ell \in [L]$) encodes the observed source sequence S_ℓ^k and past received output symbols Y^{j-1} into a symbol $X_{\ell j}$ and transmits it over the shared channel

$$Y = \sum_{\ell=1}^{L} X_\ell + Z,$$

where $Z \sim \mathcal{N}(0,1)$ and $X_\ell \in \mathbb{R}$ for all $\ell \in [L]$. Additionally, each encoder (indexed by $\ell \in [L]$) needs to satisfy the average power constraint

$$\frac{1}{n}\sum_{j=1}^{n} x_{\ell j}^2 \le P,$$

for some fixed $P > 0$. The fusion center wishes to recover an element-wise function $f(s_1, \cdots, s_L)$ losslessly from the received sequence Y^n.

A (k, n) block code for function computation over the Gaussian MAC with feedback consists of

- L encoders, where Encoder $\ell \in [L]$ assigns a symbol $x_{\ell j}(s_\ell^k, y^{j-1}) \in \mathcal{X}_\ell$ to each tuple $(s_\ell^k, y^{j-1}) \in [d]_{-1}^k \times \mathcal{Y}^{j-1}$ for all $j \in [n]$;
- one decoder, which assigns an estimate $\hat{w}^k$ to each sequence $y^n \in \mathcal{Y}^n$.

We say that the computation rate $R := k/n$ is achievable if there exists a sequence of (nR, n) computation codes such that the probability of error

$$P_e^{(n)} := \mathbb{P}\left(\bigcup_{i\in[nR]} \left\{\hat{W}_i \neq f(S_{1i}, \cdots, S_{Li})\right\}\right)$$

converges to zero as n tends to infinity. Note that the computation rate is the number of reliably computed functions per channel use. Finally, the computation capacity C is the supremum over all achievable computation rates.

Next, we provide the defintion of type-threshold functions, which are the focus of this chapter. Let Λ be a finite alphabet. Let $\{f_L\}_{L\in\mathbb{Z}^+}$ be a sequence of symmetric functions for which there exists a function $g : \mathbb{N}^d \to \Lambda$ such that for all $L \in \mathbb{Z}^+$, if $f_L(s_1, s_2, \cdots, s_L) = g(b_0, b_1, \cdots, b_{d-1})$, then

$$f_{L+1}(s_1, s_2, \cdots, s_L, s_{L+1}) = g(b'_0, b'_1, \cdots, b'_{d-1}),$$

where $b'_\ell = b_\ell + \mathbb{1}\{s_{L+1} = \ell\}$, $\ell \in [d]_{-1}$.

Definition 7.1 (Type-Threshold Function)**.** *We say that the sequence $\{f_L\}_{L\in\mathbb{Z}^+}$ belongs to the class of type-threshold functions if there exists a non-negative integer vector $\theta_{[d]_{-1}}$ and a function $g : [\theta_0]_{-1} \times [\theta_1]_{-1} \times \cdots \times [\theta_{d-1}]_{-1} \to \Lambda$ such that for all $L \in \mathbb{Z}^+$,*

$$f_L(s_1, s_2, \cdots, s_L) = g(\bar{b}_0, \cdots, \bar{b}_{d-1}),$$

where $\bar{b}_\ell := \min\{\theta_\ell, b_\ell\}$ for all $\ell \in [d]_{-1}$. The vector $\theta_{[d]_{-1}}$ is called threshold vector and $\bar{b}_\ell$ is called clipped frequency of ℓ. In the sequel, we will simply write f and the number of arguments L will be clear from context.

Some common instances of type-threshold functions are

1. the maximum, with a threshold vector $(0, 1, \cdots, 1)$;
2. the number of distinct elements, with a threshold vector $(1, 1, \cdots, 1)$;
3. the average of the m largest values $\max\limits_{\mathcal{S}\subseteq[L]:|S|=m} \dfrac{1}{\ell}\sum\limits_{\ell\in\mathcal{S}} s_\ell$, with a threshold vector $(0, m, \cdots, m)$;
4. the frequency indicator $\mathbb{1}\{\exists \ell \in [L] \text{ s.t. } s_\ell = m\}$, for which a threshold vector is the standard unit vector with 1 on the m-th position;
5. the list of heavy hitters $\{\ell \in [d]_{-1} | b_\ell \geq T\}$, with a threshold vector $(T, T, \cdots, T)$.

Note that while the average of the m largest values is a type-threshold function, the average $\frac{1}{L}\sum_{\ell=1}^{L} s_\ell$ is not.

7.2 Exploiting Interaction: Descriptions of the Clipped Frequencies

We first demonstrate how to exploit interaction through noiseless causal feedback. One important benefit of interaction is that every sensor is aware of the status of the fusion center's knowledge of the desired function. We borrow some ideas from the framework of interactive source coding [54]. Fix $N \in \mathbb{Z}^+$ and a mapping $\kappa : [N] \to [L]$, where $N \geq L$. The basic idea in the achievability of interactive source coding is as follows. We divide the whole communication into N rounds. In each round (indexed by $\ell \in [N]$), only sensor node $\kappa(\ell)$ is activated. The activated sensor $\kappa(\ell)$ quantizes the source vector $\mathbf{s}_{\kappa(\ell)}$ into a vector $\mathbf{v}^{(\ell)}$ with side information $(\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(\ell-1)})$ received in previous rounds and then broadcasts this common description $\mathbf{v}^{(\ell)}$ to all other nodes in the network. After N rounds, the fusion center computes the desired function based on the received N descriptions.

Thanks to the arithmetic computation coding developed in Section 6.3, the sensors can cooperatively form one single description using linear combination. Intuitively, we want to use the superposition property to somehow *merge* the descriptions so that the amount of information received at the fusion center is reduced but still sufficient to deduce the desired function. A simple first attempt is to consider general descriptions: Consider the descriptions $\{V_m^{(\ell)}\}_{m\in[L],\ell\in[N]}$ satisfying

1. $H(f(S_1, \cdots, S_L)|U_{[N]}) = 0$,
2. $V_m^{(\ell)} \multimap (U_{[\ell-1]}, S_m) \multimap S_{[L]\setminus\{m\}}$ form a Markov chain,

where

$$U_\ell = \sum_{m=1}^{L} V_m^{(\ell)},$$

in which the superposition is embedded. These descriptions are very general but seem hard to analyze. Instead, we next propose a more constrained set of auxiliary random variables (descriptions). Not only can these descriptions be analyzed, they also have a natural operational meaning.

Rather than generating descriptions directly for the desired type-threshold function, we construct descriptions for the clipped frequencies. The reasons are twofold. First, the clipped frequencies contain all the information needed to deduce the desired type-threshold function. Second, as can be seen in (6.6), the clipped frequencies are sums of indicators with a clipping. Thus, the indicators can serve as descriptions and the addition can play the role of merge, which is naturally matched with the superposition property of the Gaussian MAC. In order to reduce the entropy of the descriptions, it might be unwise to attain the whole frequency and then do the clipping. Instead, we consider a recursive approach: Update only a partial sum of indicators and perform the clipping on a regular basis. Now come the details.

First, for each $\ell \in [d]_{-1}$, we attribute a composition of $[L]$: $\mathcal{A}_1^{(\ell)}, \cdots, \mathcal{A}_{J_\ell}^{(\ell)}$, which satisfies that 1) $\mathcal{A}_j^{(\ell)} \neq \emptyset$ for all j, 2) $\bigcup_j \mathcal{A}_j^{(\ell)} = [L]$, 3) $\mathcal{A}_i^{(\ell)} \bigcap \mathcal{A}_j^{(\ell)} = \emptyset$ for all $i \neq j$. Then, the sensors whose index lies in the same subset, say, $\mathcal{A}_m^{(\ell)}$, form a group.

Note that the formation of the groups can be different for each ℓ. Each group is responsible for a partial sum of indicators.

Denote by $U_1^{(\ell)}, U_2^{(\ell)}, \cdots$ the descriptions of the clipped frequency $\overline{B}_\ell$, $\ell \in [d]_{-1}$. Then, the descriptions of the clipped frequency $\overline{B}_\ell$ is defined by the following recursion

$$U_m^{(\ell)} = U_{m-1}^{(\ell)} + \sum_{i \in \mathcal{A}_m^{(\ell)}} \mathbb{1}\{U_{m-1}^{(\ell)} < \theta_\ell\} \bigcap \{S_i = \ell\}, \tag{7.2}$$

for all $m \in [J_\ell]$, where $U_0^{(\ell)} = 0$. As can be seen, $\sum_{i \in \mathcal{A}_m^{(\ell)}} \mathbb{1}\{S_i = \ell\}$ is the partial sum of indicators just mentioned and the event $\{U_{m-1}^{(\ell)} < \theta_\ell\}$ plays the role of clipping. Note that $U_{[J_\ell]}^{(\ell)}$ are random variables induced by the sources $S_{[L]}$. It is clear that the clipped frequency $\overline{B}_\ell$ is equal to $\min\{U_{J_\ell}^{(\ell)}, \theta_\ell\}$ and thus the fusion center can deduce the desired function once it learns all descriptions $\left(U_{[J_0]}^{(0)}, U_{[J_1]}^{(1)}, \cdots, U_{[J_{d-1}]}^{(d-1)}\right)$.

7.2.1 Entropy of Descriptions

As will be clear in Section 7.3, the entropy of $\left(U_{[J_0]}^{(0)}, U_{[J_1]}^{(1)}, \cdots, U_{[J_{d-1}]}^{(d-1)}\right)$ determines the achievable computation rate of the proposed scheme and we want this entropy to be as small as possible. In particular, we are interested in how this entropy scales as the number of sensors increases since it directly affects the scaling law of the achievable computation rate. Since the entropy of the descriptions is governed by the chosen compositions, the goal is to characterize a pattern of compositions which results in a bounded entropy as the number of sensors increases. For this, we will consider different distribution ensembles, which are families of probability distributions $\{\prod_{m=1}^{L} p_{S_m}\}_{L \in \mathbb{Z}^+}$. Let us start with an example.

Example 7.1 (Binary Maximum)**.** *Assume that $S_m \in \{0, 1\}$ for all $m \in [L]$. The binary maximum is defined as $S_{\max} := \max S_{[L]}$. Intuitively, if we know that one sensor observes a value of one, then the function value can already be determined even though the observations at other sensors are unknown. Note that $(\theta_0, \theta_1) = (0, 1)$ is a valid threshold vector of the binary maximum and thus $U_m^{(0)} = 0$ for all m.*

Let us consider independent and identically distributed (i.i.d.) ensembles with Bernoulli(β), where $0 < \beta < 1$ and β might depend on L. For convenience, we use the term size-a composition, where $a \in [L]$, to refer to any composition satisfying $|\mathcal{A}_j| = a$ for all $j \in [J-1]$ and $|\mathcal{A}_J| = L - (J-1)a$, where $J = \lfloor L/a \rfloor$. The entropy of the descriptions $U_{[J_1]}^{(1)}$ under size-a compositions can be evaluated as

$$\begin{aligned} H\left(U_{[J_1]}^{(1)}\right) &\overset{(a)}{=} \sum_{m=1}^{J_1} H\left(U_m^{(1)} \middle| U_{m-1}^{(1)}\right) \\ &\overset{(b)}{=} \sum_{m=1}^{J_1} \mathbb{P}\left(U_{m-1}^{(1)} = 0\right) H\left(U_m^{(1)} \middle| U_{m-1}^{(1)} = 0\right) \\ &\overset{(c)}{=} \frac{1-(1-\beta)^{(J_1-1)a}}{1-(1-\beta)^a} H(Q_a) + (1-\beta)^{(J_1-1)a} H(Q_{L-(J_1-1)a}), \end{aligned} \tag{7.3}$$

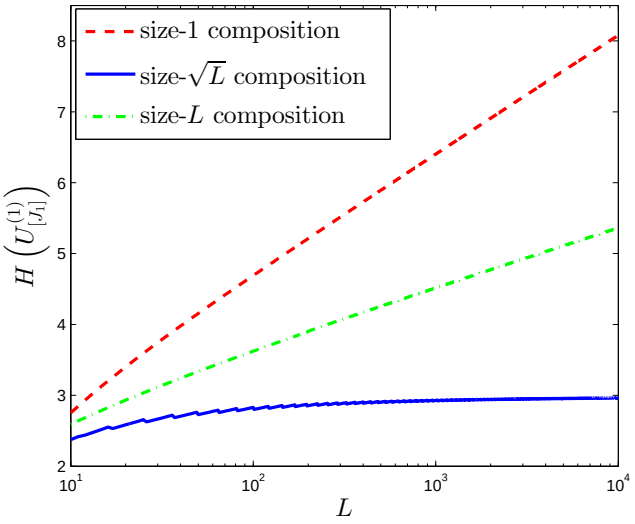

Figure 7.2: The entropy of the descriptions $U^{(1)}_{[J_1]}$ (Expression (7.3)) under various compositions for the i.i.d. source ensemble in which each source follows Bernoulli$\left(\frac{1}{\sqrt{L}}\right)$, where size-$a$ composition is the composition satisfying that $|\mathcal{A}_j| = a$ for all $j \in [J_1 - 1]$.

where $Q_m \sim$ *Binomial*(m, β), (a) *follows from the independence of* $S_{[L]}$, (b) *follows since* $U^{(1)}_m$ *conditioned on* $\{U^{(1)}_{m-1} \geq 1\}$ *is deterministic, and* (c) *follows since*

$$\mathbb{P}\left(U^{(1)}_{m-1} = 0\right) = (1-\beta)^{(m-1)a},$$

$$H\left(U^{(1)}_m \middle| U^{(1)}_{m-1} = 0\right) = H\left(\sum_{i \in \mathcal{A}^{(\ell)}_m} \mathbb{1}\{S_i = 1\}\right),$$

and $\sum_{i \in \mathcal{A}^{(\ell)}_m} \mathbb{1}\{S_i = 1\} \sim$ *Binomial*$(|\mathcal{A}^{(\ell)}_m|, \beta)$.

Now we discuss the following three cases.

1) i.i.d. sources each of which follows Bernoulli(c)*, where* $c \in (0,1)$ *is a constant independent of* L

If we fix $a = 1$*, then* (7.3) *becomes*

$$H\left(U^{(1)}_{[J_1]}\right) = \frac{1}{c}(1-(1-c)^{(L-1)})h_2(c) \underset{L\to\infty}{\longrightarrow} \frac{h_2(c)}{c}.$$

For this ensemble, the simple one-at-a-time approach gives a bounded entropy of descriptions as L *increases. By contrast, if we substitute* $a = L$ *into* (7.3)*, then* $H\left(U^{(1)}_{[J_1]}\right) = H(Q_L) = \Theta(\log L)$*. Thus, the size-*$L$ *composition fails to achieve a bounded entropy of descriptions.*

2) i.i.d. sources each of which follows Bernoulli$\left(\frac{1}{L}\right)$

If we fix $a = L$*, then* (7.3) *becomes*

$$H\left(U^{(1)}_{[J_1]}\right) = H(Q_L) \overset{(a)}{\leq} \frac{1}{2}\log\left(2\pi e\left(1+\frac{1}{12}\right)\right) \approx 2.1,$$

where (a) follows from [55, Theorems 7 and 8] and [56, Expression (1)]. Thus, for this ensemble, the size-L composition achieves a bounded entropy of descriptions as L increases. By contrast, if the size-1 composition is applied, then

$$H\left(U^{(1)}_{[J_1]}\right) = L\left(1-\left(1-\frac{1}{L}\right)^{(L-1)}\right) h_2\left(\frac{1}{L}\right) \geq \frac{1}{2}\log L.$$

Thus, the size-1 composition fails to achieve a bounded entropy of descriptions.
3) i.i.d. sources each of which follows Bernoulli$\left(\frac{1}{\sqrt{L}}\right)$
Figure 7.2 plots $H\left(U^{(1)}_{[J_1]}\right)$ for the size-1 composition, the size-$\sqrt{L}$ composition, and the size-L composition. As can be seen, as L increases, only the size-$\sqrt{L}$ composition achieves a bounded entropy of descriptions, which will be proved in Lemma 7.1.

As shown in the above example, different distribution ensembles require different compositions to achieve a bounded entropy of descriptions. The following lemma shows the existence of compositions that guarantee a bounded entropy of descriptions for any type-threshold function when the sources are independent.

Lemma 7.1. *Fix a threshold vector $\theta_{[d]_{-1}}$ and a joint pmf $\prod_{m=1}^{L} p_{S_m}$. For each $\ell \in [d]_{-1}$, there exists a composition such that*

$$H\left(U^{(\ell)}_{[J_\ell]}\right) < \frac{5}{2}\log(1+\theta_\ell) + 12.$$

Proof: We refer to Appendix for the proof. ∎

Using Lemma 7.1, we can upper bound the entropy of descriptions achieved by the optimum compositions as

$$H\left(U^{(0)}_{[J_0]}, U^{(1)}_{[J_1]}, \cdots, U^{(d-1)}_{[J_{d-1}]}\right) \leq 12d + \frac{5}{2}\sum_{\ell=0}^{d-1}\log(1+\theta_\ell), \tag{7.4}$$

which is independent of the number of sensors.

7.2.2 Tailoring to the Maximum Function

The descriptions introduced in (7.2) are a general framework for every type-threshold function. However, for many functions, it is unnecessary to acquire specific values of all clipped frequencies so as to deduce the function value. The simplest example is the frequency indicators for which we only care about one single frequency. Yet another example is the maximum function. If we directly use (7.2), then we need to convey $(d-1)$ clipped frequencies and the entropy of their descriptions is upper bounded by $\Theta(d)$ as shown in (7.4). However, once all nodes learn that $\bar{b}_\ell = 1$, then the values of $\bar{b}_{[\ell]_{-1}}$ are irrelevant since the maximum must be larger than or equal to ℓ.

In this subsection, we consider an adaptation of the descriptions for the maximum function based on the *binary search* algorithm. Fix $\lceil \log d \rceil$ compositions: $\mathcal{A}^{(\ell)}_{[J_\ell]}$, $\ell \in [\lceil \log d \rceil]$. For each ℓ, define the recursion

$$\tilde{U}^{(\ell)}_m = \tilde{U}^{(\ell)}_{m-1} + \sum_{i \in \mathcal{A}^{(\ell)}_m} \mathbb{1}\{\tilde{U}^{(\ell)}_{m-1} = 0\}\mathbb{1}\{S_i \geq D_\ell\},$$

where

$$D_\ell = \left\lceil \frac{d}{2^\ell} \left(1 + \sum_{j=1}^{\ell-1} \mathbb{1}\{\tilde{U}_{J_j}^{(j)} > 0\} 2^{\ell-j} \right) \right\rceil$$

is the midpoint in the ℓ-th stage of the binary search. For example, $D_1 = \lceil \frac{d}{2} \rceil$, $D_2 \in \{\lceil \frac{d}{4} \rceil, \lceil \frac{3d}{4} \rceil\}$, $D_3 \in \{\lceil \frac{d}{8} \rceil, \lceil \frac{3d}{8} \rceil, \lceil \frac{5d}{8} \rceil, \lceil \frac{7d}{8} \rceil\}$, and so on. Note that $\min\{D_{\lceil \log d \rceil}, d\} = \max S_{[L]}$. Therefore, the fusion center can deduce the maximum once it learns the sequence $\left(\tilde{U}_{[J_1]}^{(1)}, \tilde{U}_{[J_2]}^{(2)}, \cdots, \tilde{U}_{[J_{\lceil \log d \rceil}]}^{(\lceil \log d \rceil)}\right)$. Since the proof of Lemma 7.1 still follows after replacing $\mathbb{P}(S_i = \ell)$ by $\mathbb{P}(S_i \geq D_\ell)$ and substituting $\theta_\ell = 1$, the entropy $H\left(\tilde{U}_{[J_\ell]}^{(\ell)}\right)$ can be upper bounded by a constant. Thus, the entropy of the descriptions for the maximum function is reduced from $\Theta(d)$ to $\Theta(\log d)$.

7.3 Multi-Round Group Broadcast

In this section, we elaborate the developed coding scheme termed *multi-round group broadcast*. In brief, the multi-round group broadcast conveys the descriptions of clipped frequencies introduced in Section 7.2 over the Gaussian MAC with feedback. Thanks to the feedback, all sensors can also decode the descriptions of clipped frequencies. Before going into the details, we first give a high-level overview. To explain the main idea, it suffices to consider one of the clipped frequencies ℓ. Let the threshold θ_ℓ and the composition $\mathcal{A}_{[J_\ell]}^{(\ell)}$ be fixed. The operations given below are performed symbol-wise.

Before transmission, each node sets up a counter with initial value zero. There are totally J_ℓ rounds. In round $m \in [J_\ell]$, all nodes in $\mathcal{A}_m^{(\ell)}$ are activated and broadcast the indicator "ℓ is observed". The transmitted indicators are superimposed by the channel. Then, every node decodes the arithmetic sum of the indicators and increment the counter by the corresponding value. If the value of every counter reaches or exceeds the threshold θ_ℓ, then every node learns the clipped frequency $\overline{b}_\ell = \theta_\ell$ and we can jump directly to the next frequency; otherwise, we move on to the next round.

Since the power constraint is imposed as an average over long time horizons, each group can use larger transmit power during its active time period by accumulating power in its non-active time period. Now we present our main theorem.

Theorem 7.1. *Consider computation of a type-threshold function with threshold vector $\theta_{[d]_{-1}}$ over the Gaussian MAC with noiseless causal feedback. Fix a composition $\mathcal{A}_{[J_\ell]}^{(\ell)}$ for each $\ell \in [d]_{-1}$. Then, for any values $\alpha_m^{(\ell)} \geq 0$ and $P_m^{(\ell)} \geq 0$ satisfying $\sum_{\ell=0}^{d-1} \sum_{m=1}^{J_\ell} \alpha_m^{(\ell)} \leq 1$ and $\sum_{(\ell,m) \text{ s.t. } i \in \mathcal{A}_m^{(\ell)}} \alpha_m^{(\ell)} P_m^{(\ell)} \leq P$ for all $i \in [L]$, any computation rate R satisfying*

$$R < \min_{\ell \in [d]_{-1}} \min_{m \in [J_\ell]} \frac{\frac{\alpha_m^{(\ell)}}{2} \log^+ \left(\frac{1}{|\mathcal{A}_m^{(\ell)}|} + P_m^{(\ell)} \right)}{H\left(U_{m_{(Q_\ell)}}^{(\ell)} \middle| U_{[J_0]}^{(0)}, U_{[J_1]}^{(1)}, \cdots, U_{[m_{(Q_\ell)}-1]}^{(\ell)}, Q_{[d]_{-1}} \right)} \tag{7.5}$$

is achievable, where

$$U^{(\ell)}_{m_{(Q_\ell)}} = U^{(\ell)}_{m_{(Q_\ell)}-1} + \sum_{i\in\mathcal{A}^{(\ell)}_m} \mathbb{1}\{U^{(\ell)}_{m_{(Q_\ell)}-1} < \theta_\ell\}\mathbb{1}\{S_i = \ell\}, \tag{7.6}$$

$U^{(\ell)}_0 = 0$, $m_{(Q_\ell)} := ((m + Q_\ell - 1) \mod J_\ell) + 1$, *and* $Q_\ell \sim$ *Uniform*$([J_\ell]_{-1})$.

Proof: Assuming that the descriptions $(\mathbf{u}^{(0)}_{[J_0]}, \mathbf{u}^{(1)}_{[J_1]}, \cdots, \mathbf{u}^{(\ell)}_{[m-1]})$ are successfully decoded at all nodes, we consider the transmission of $\mathbf{u}^{(\ell)}_m$ by the group $\mathcal{A}^{(\ell)}_m$. Denote by $R^{(\ell)}_m$ and $P^{(\ell)}_m$ the computation rate and the transmit power of $\mathbf{u}^{(\ell)}_m$, respectively. Note that we can treat $(S_i, U^{(0)}_{[J_0]}, U^{(1)}_{[J_1]}, \cdots, U^{(\ell)}_{[m-1]})$ as an augmented source observed at Sensor $i \in [L]$. Then, applying Theorem 6.5 and Expression (7.1), we have that any computation rate $R^{(\ell)}_m$ satisfying

$$\begin{aligned} R^{(\ell)}_m &< \frac{\frac{1}{2}\log^+\left(\frac{1}{|\mathcal{A}^{(\ell)}_m|} + P^{(\ell)}_m\right)}{H\left(\sum_{i\in\mathcal{A}^{(\ell)}_m} \mathbb{1}\{U^{(\ell)}_{m-1} < \theta_\ell\}\mathbb{1}\{S_i = \ell\}\middle| U^{(0)}_{[J_0]}, U^{(1)}_{[J_1]}, \cdots, U^{(\ell)}_{[m-1]}\right)} \\ &= \frac{\frac{1}{2}\log^+\left(\frac{1}{|\mathcal{A}^{(\ell)}_m|} + P^{(\ell)}_m\right)}{H\left(U^{(\ell)}_m\middle| U^{(0)}_{[J_0]}, U^{(1)}_{[J_1]}, \cdots, U^{(\ell)}_{[m-1]}\right)} \end{aligned}$$

is achievable, where

$$U^{(\ell)}_m = U^{(\ell)}_{m-1} + \sum_{i\in\mathcal{A}^{(\ell)}_m} \mathbb{1}\{U^{(\ell)}_{m-1} < \theta_\ell\}\mathbb{1}\{S_i = \ell\}.$$

For each $\ell \in [d]_{-1}$, if the transmission order remains fixed, then the first group will consume more power than the last group since many times the threshold has already been reached before the last group has a chance to transmit. To avoid such unbalanced situation, we apply a block-wise time sharing among different rotations, i.e., each group takes turns being the first group to transmit. Therefore, with $U^{(\ell)}_{m_{(Q_\ell)}}$ defined as in (7.6), any computation rate $R^{(\ell)}_m$ smaller than

$$\frac{\frac{1}{2}\log^+\left(\frac{1}{|\mathcal{A}^{(\ell)}_m|} + P^{(\ell)}_m\right)}{H\left(U^{(\ell)}_{m_{(Q_\ell)}}\middle| U^{(0)}_{[J_0]}, U^{(1)}_{[J_1]}, \cdots, U^{(\ell)}_{[m_{(Q_\ell)}-1]}, Q_{[d]_{-1}}\right)}$$

is achievable, where $m_{(Q_\ell)}$ is the transmission order of the group $\mathcal{A}^{(\ell)}_m$.

Finally, to satisfy the power constraint, we attribute $\alpha^{(\ell)}_m$ fraction of time to each round satisfying

$$\begin{aligned} \sum_{\ell=0}^{d-1}\sum_{m=1}^{J_\ell} \alpha^{(\ell)}_m &\le 1, \\ \sum_{(\ell,m) \text{ s.t. } i\in\mathcal{A}^{(\ell)}_m} \alpha^{(\ell)}_m P^{(\ell)}_m &\le P \text{ for all } i \in [L], \end{aligned}$$

and thus any computation rate R satisfying (7.5) is achievable, which establishes the theorem. ∎

We remark that Expression (7.6) is equivalent to saying that

$$U_m^{(\ell)} = U_{m-1}^{(\ell)} + \sum_{i \in \mathcal{A}_{m-(L-Q_\ell)}^{(\ell)}} \mathbb{1}\{U_{m-1}^{(\ell)} < \theta_\ell\} \mathbb{1}\{S_i = \ell\}.$$

Next, we would like to gain insight into (7.5). The denominator of (7.5) can be upper bounded as

$$\begin{aligned}
& H\left(U_{m_{(Q_\ell)}}^{(\ell)} \middle| U_{[J_0]}^{(0)}, U_{[J_1]}^{(1)}, \cdots, U_{[m_{(Q_\ell)}-1]}^{(\ell)}, Q_{[d]_{-1}}\right) \\
\le\ & H\left(U_{m_{(Q_\ell)}}^{(\ell)} \middle| U_{m_{(Q_\ell)}-1}^{(\ell)}, Q_\ell\right) \\
=\ & \frac{1}{J_\ell} \sum_{q=0}^{J_\ell - 1} H\left(U_{m_{(q)}}^{(\ell)} \middle| U_{m_{(q)}-1}^{(\ell)}, Q_\ell = q\right) \\
=\ & \frac{1}{J_\ell} \sum_{q=0}^{J_\ell - 1} H\left(U_{m_{(q)}}^{(\ell)} \middle| U_{m_{(q)}-1}^{(\ell)}\right). \qquad (7.7)
\end{aligned}$$

The following lemma shows the existence of compositions that guarantee (7.7) is bounded. Note that Lemma 7.1 is a special case of Lemma 7.2 with $q = 0$.

Lemma 7.2. *Fix a threshold vector $\theta_{[d]_{-1}}$ and a joint pmf $\prod_{m=1}^{L} p_{S_m}$. For each $\ell \in [d]_{-1}$, there exists a composition such that for any $q \in [J_\ell]_{-1}$,*

$$\sum_{m=0}^{J_\ell - 1} H\left(U_{m_{(q)}}^{(\ell)} \middle| U_{m_{(q)}-1}^{(\ell)}\right) < \frac{5}{2} \log(1 + \theta_\ell) + 12, \qquad (7.8)$$

where $U_{m_{(q)}}^{(\ell)}$ is given by (7.6) with $Q_\ell = q$.

Proof: We refer to Appendix for the proof. ∎

If we use Lemma 7.2 to lower bound the rate given in Theorem 7.1 (Expression (7.5)), we obtain the following corollary:

Corollary 7.1. *Consider computation of a type-threshold function with threshold vector $\theta_{[d]_{-1}}$ over the Gaussian MAC with noiseless causal feedback. Any computation rate R satisfying*

$$R < \max_{\beta \in (0,1]} \frac{\frac{\beta}{2} \log^+\left(\frac{1}{L} + \frac{\min_{\ell \in [d]_{-1}} J_\ell}{\beta} P\right)}{12d + \frac{5}{2} \sum_{\ell=0}^{d-1} \log(1 + \theta_\ell)}$$

is achievable by the multi-round group broadcast, where $J_{[d]_{-1}}$ are determined by the composition used in the proof of Lemma 7.2.

Proof: First, combining Theorem 7.1, Expression (7.7), and Lemma 7.2 shows that any computation rate R satisfying

$$R < \min_{\ell \in [d]_{-1}} \min_{m \in [J_\ell]} \frac{\frac{\alpha_m^{(\ell)}}{2} \log^+\left(\frac{1}{|\mathcal{A}_m^{(\ell)}|} + P_m^{(\ell)}\right)}{\frac{1}{J_\ell}\left(\frac{5}{2} \log(1 + \theta_\ell) + 12\right)}$$

is achievable. Then, setting $\alpha_m^{(\ell)} = \beta\alpha_\ell/J_\ell$, $P_m^{(\ell)} = J_\ell P/\beta$ and noticing $|\mathcal{A}_m^{(\ell)}| \le L$ gives

$$R < \min_{\ell\in[d]_{-1}} \frac{\frac{\beta\alpha_\ell}{2}\log^+\left(\frac{1}{L}+\frac{J_\ell}{\beta}P\right)}{\frac{5}{2}\log(1+\theta_\ell)+12},$$

where $\alpha_\ell \ge 0$, $\sum_{\ell=0}^{d-1}\alpha_\ell \le 1$, and $\beta \in (0,1]$. Finally, we set

$$\alpha_\ell = \frac{\frac{5}{2}\log(1+\theta_\ell)+12}{12d+\frac{5}{2}\sum_{i=0}^{d-1}\log(1+\theta_i)},$$

and thus we have

$$R < \frac{\frac{\beta}{2}\log^+\left(\frac{1}{L}+\frac{\min_{\ell\in[d]_{-1}} J_\ell}{\beta}P\right)}{12d+\frac{5}{2}\sum_{i=0}^{d-1}\log(1+\theta_i)}. \tag{7.9}$$

The corollary is established after we maximize the right hand side of (7.9) over $\beta \in (0,1]$. ■

Corollary 7.1 establishes two key facts. First, even if the number of sensors tends to infinity, the multi-round group broadcast still achieves a positive rate as long as $P > 0$, which establishes the worst-case scaling law $\Omega(\log P)$ (see Table 7.1). Second, depending on the source distribution, the achievable computation rate can even increase with the number of sensors through the gain $\min_{\ell\in[d]_{-1}} J_\ell$.

7.3.1 Scaling Law for the Number of Sensors and the Transmit Power: Binary Maximum

In this subsection, we study the interplay among the number of sensors and the transmit power over the Gaussian MAC with feedback. We consider the binary maximum function introduced in Section 7.2.1. For the binary maximum, the rate expression (7.5) can be simplified as

$$R < \min_{m\in[J]} \frac{\frac{1}{2}\log^+\left(\frac{1}{|\mathcal{A}_m^{(1)}|}+JP\right)}{\sum_{q=0}^{J-1} H\left(U_{m_{(q)}}^{(1)}\middle|U_{m_{(q)}-1}^{(1)}\right)}. \tag{7.10}$$

Again, we consider the following three distribution ensembles.
1) i.i.d. sources each of which follows Bernoulli(c), where $c \in (0,1)$ is a constant independent of L
For this ensemble, the multi-round group broadcast with the size-1 composition achieves the scaling law of $\Theta(\log LP)$. We remark that applying interactive source coding in [54] can achieve a higher computation rate but within the same scaling law.
2) i.i.d. sources each of which follows Bernoulli$\left(\frac{1}{L}\right)$
For this ensemble, the size-1 composition achieves the scaling law of $\Theta\left(\frac{\log LP}{\log L}\right)$. By contrast, the size-$L$ composition achieves the scaling law of $\Theta(\log P)$. Depending on the available power, there is a trade-off between power accumulation and reduction of the entropy of descriptions.

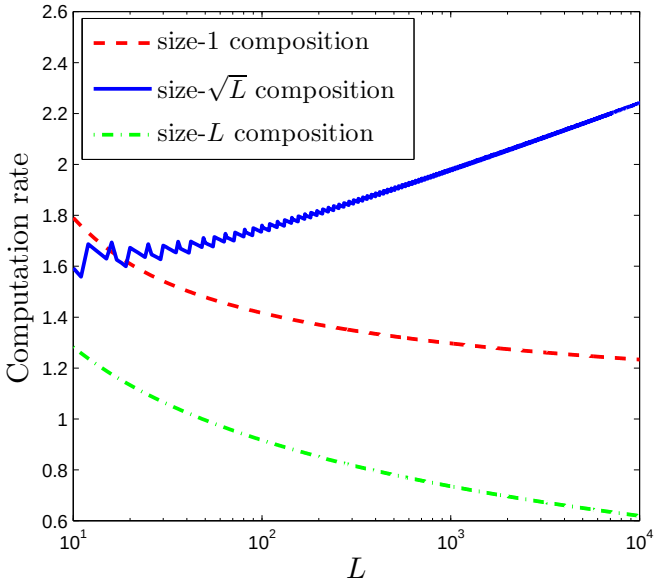

Figure 7.3: Evaluation of (7.10), with $P = 20$ dB, for the achievable computation rates of the binary maximum function for the i.i.d. source ensemble in which each source follows Bernoulli$\left(\frac{1}{\sqrt{L}}\right)$.

3) i.i.d. sources each of which follows Bernoulli$\left(\frac{1}{\sqrt{L}}\right)$

Figure 7.3 plots the computation rates of the proposed multi-round group broadcast with the size-1, size-$\sqrt{L}$, and size-L compositions at $P = 20$ dB. The figure shows that as L increases, the achievable computation rate of the size-$\sqrt{L}$ composition grows logarithmically with L, while the other two compositions achieve at most a constant rate. The reason that the computation rate can increase with L is that, for this ensemble the size-$\sqrt{L}$ composition satisfies the upper bound (7.8) and at the same time each group can accumulate power in its non-active time period, roughly $(1 - 1/\sqrt{L})$ fraction of time.

7.4 Upper Bound

In this section, we provide a simple cut-set based upper bound on the computation capacity for arbitrary functions over the Gaussian MAC with feedback. In general, the derived upper bound can not be matched by the achievabilities presented in this paper. We remark that it might be possible to tighten the upper bound by applying the converse of interactive source coding for function computation [57].

Let $\Omega \subseteq [L]$ and $\Omega^c := [L] \backslash \Omega$. First, assume that a genie provides $\mathbf{s}_{\Omega^c}$ to every node. Given $\mathbf{s}_{\Omega^c}$ as side information, the minimum source coding rate for computing the function f should be at least $H(f(S_1, \cdots, S_L)|S_{\Omega^c})$. Second, we treat sensor nodes in Ω as a supernode-Ω to which $\mathbf{s}_\Omega$ are available. Also, we treat sensor nodes in Ω^c and the fusion center as supernode-$\{0\} \bigcup \Omega^c$. Thus, the channel from the supernode-Ω to the supernode-$\{0\} \bigcup \Omega^c$ is a point-to-point multiple-input multiple-output (MIMO) channel in which the source–channel separation theorem holds and

feedback does not increase the capacity since the channel is memoryless. Therefore, following similar lines in the proof of the cut-set bound [58, Theorem 15.10.1], the computation capacity of the function f is upper bounded as

$$C \leq \max_{p_{X_{[L]}}} \min_{\substack{\Omega \subseteq [L] \\ \Omega \neq \emptyset}} \frac{I(X_\Omega; Y_0, Y_{\Omega^c} | X_{\Omega^c})}{H(f(S_1, \cdots, S_L)|S_{\Omega^c})}, \tag{7.11}$$

where the input distribution $p_{X_{[L]}}$ subjects to some input cost constraints. After applying the Gaussian assumption and the discretization procedure (see Section 3.4.1 in [5]) to (7.11), we have the following proposition.

Proposition 7.1. *Consider computation of an arbitrary function f over the Gaussian MAC with noiseless causal feedback. The computation capacity C is upper bounded as*

$$C \leq \max_{\mathbf{K}} \min_{\substack{\Omega \subseteq [L] \\ \Omega \neq \emptyset}} \frac{\frac{1}{2} \log \left(1 + (L + 1 - |\Omega|) \sum_{i,j} [\mathbf{K}_{X_\Omega | X_{\Omega^c}}]_{ij}\right)}{H(f(S_1, \cdots, S_L)|S_{\Omega^c})}, \tag{7.12}$$

where the matrix $\mathbf{K}$ is positive semidefinite with the (i,i) entry $[\mathbf{K}]_{ii} \leq P$, $i \in [L]$ and $\mathbf{K}_{X_\Omega | X_{\Omega^c}}$ is the conditional covariance matrix of X_Ω given X_{Ω^c} for $X_{[L]} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$.

Proof: Denote by $\mathbf{K}$ the covariance matrix of $X_{[L]}$ with the (i,i) entry $[\mathbf{K}]_{ii} \leq P$, $i \in [L]$. Applying Theorem 19.1 in [5], the mutual information term in (7.11) can be upper bounded as

$$\begin{aligned} I(X_\Omega; Y_{\Omega^c} | X_{\Omega^c}) &\leq \frac{1}{2} \log \left(\det \left(\mathbf{I} + \mathbf{G} \mathbf{K}_{X_\Omega | X_{\Omega^c}} \mathbf{G}^T\right)\right) \\ &= \frac{1}{2} \log \left(1 + (L + 1 - |\Omega|) \sum_{i,j} [\mathbf{K}_{X_\Omega | X_{\Omega^c}}]_{ij}\right), \end{aligned}$$

where $\mathbf{I}$ is the $(L+1-|\Omega|) \times (L+1-|\Omega|)$ identity matrix, $\mathbf{G}$ is the $(L+1-|\Omega|) \times |\Omega|$ all-one matrix, and $\mathbf{K}_{X_\Omega | X_{\Omega^c}}$ is the conditional covariance matrix of X_Ω given X_{Ω^c}. The equality holds if $X_{[L]} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$. Then, the proposition follows immediately. ■

Example 7.2. *Consider computation of the binary maximum over the Gaussian MAC with feedback. Assume that the distribution ensemble is i.i.d. Bernoulli$\left(\frac{1}{L}\right)$. Recall that the multi-round group broadcast with size-1 composition achieves the scaling law of $\Theta\left(\frac{\log LP}{\log L}\right)$, whereas the multi-round group broadcast with size-L composition achieves the scaling law of $\Theta(\log P)$. If we just consider the cut $\Omega = [L]$, then the cut-set bound (7.12) can be simplified as*

$$C \leq \frac{\frac{1}{2} \log(1 + L^2 P)}{h_2\left((1 - 1/L)^L\right)}.$$

Thus, in this ensemble the scaling of the cut-set bound (7.12) is $\Theta(\log LP)$.

Appendix: Bounded Entropy of the Descriptions of the Clipped Frequencies as $L \to \infty$

In this appendix, we provide a proof of Lemma 7.2 and then Lemma 7.1 will follow as a special case with $q = 0$. Since the proof works universally for every clipped frequency $\overline{b}_\ell$, we drop all indices ℓ in the proof for simplicity. Besides, for convenience, we denote $\beta_i := \mathbb{P}(S_i = \ell)$ for all $i \in [L]$. For the proof of Lemma 7.2, we need the following lemma, which upper bounds the entropy of the sum of independent Bernoulli random variables.

Lemma 7.3. *Fix $\beta_{[L]} \in [0,1]^L$. Let $X_{[L]}$ be independent random variables, where $X_i \sim$ Bernoulli(β_i) for all $i \in [L]$. Then,*

$$H\left(\sum_{i=1}^{L} X_i\right) \leq \frac{1}{2}\log\left(2\pi e\left(\sum_{i=1}^{L}\beta_i + \frac{1}{12}\right)\right). \tag{7.13}$$

Proof: First, applying Theorem 1 in [59], we have

$$H\left(\sum_{i=1}^{L} X_i\right) \leq H\left(\sum_{i=1}^{L}\overline{X}_i\right), \tag{7.14}$$

where $\overline{X}_{[L]}$ are i.i.d. Bernoulli$\big(\overline{\beta}\big)$ random variables and $\overline{\beta} = \frac{1}{L}\sum_{i=1}^{L}\beta_i$.

Notice that $\sum_{i=1}^{L}\overline{X}_i \sim \text{Binomial}\big(L, \overline{\beta}\big)$. Let Y be a Poisson random variable with mean $\sum_{i=1}^{L}\beta_i$. Then, we have

$$\begin{aligned} H\left(\sum_{i=1}^{L}\overline{X}_i\right) &\overset{(a)}{\leq} H(Y) \\ &\overset{(b)}{\leq} \frac{1}{2}\log\left(2\pi e\left(\sum_{i\in\mathcal{A}_m}\beta_i + \frac{1}{12}\right)\right), \end{aligned} \tag{7.15}$$

where (a) follows from [55, Theorems 7 and 8] and (b) follows from [56, Expression (1)]. Finally, combining (7.14) and (7.15), the inequality (7.13) is established. ∎

Proof of Lemma 7.2: Without loss of generality, we assume $\theta \leq L$ since there are only L sensors. If $\theta = 0$, then $U_m = 0$ for all $m \in [J]$ and thus $H\left(U_{[J]}\right) = 0$. In the following, we consider the case $1 \leq \theta \leq L$. Since the sources are independent, $U_1 \multimap U_2 \multimap \cdots \multimap U_J$ form a Markov chain. From now on, we consider a fixed $q \in [J]_{-1}$.

First, consider the case $\sum_{i=1}^{L}\beta_i \leq \theta$. In this case, we use the size-L composition. Applying Lemma 7.3 by substituting X_i with $\mathbb{1}\{S_i = \ell\}$, we have

$$\begin{aligned} H(U_1) &= H\left(\sum_{i=1}^{L}\mathbb{1}\{S_i = \ell\}\right) \\ &\leq \frac{1}{2}\log\left(2\pi e\left(\sum_{i=1}^{L}\beta_i + \frac{1}{12}\right)\right) \\ &\leq \frac{1}{2}\log\left(2\pi e\left(\theta + \frac{1}{12}\right)\right). \end{aligned}$$

Next, consider the case $\sum_{i=1}^{L} \beta_i > \theta$. Let the intervals $[a_{m-1}+1 : a_m]$, $m \in [J]$, satisfy $0 = a_0 < \cdots < a_J = L$,

$$\sum_{i=a_{m-1}+1}^{a_m-1} \beta_i < \theta \leq \sum_{i=a_{m-1}+1}^{a_m} \beta_i, \tag{7.16}$$

for $m \in [J-1]$, and

$$\theta \leq \sum_{i=a_{J-1}+1}^{L} \beta_i < 2\theta. \tag{7.17}$$

Note that for all $m \in [J-1]$, since $p_{a_m} \leq 1$, (7.16) implies that

$$\sum_{i=a_{m-1}+1}^{a_m} \beta_i < \theta + 1. \tag{7.18}$$

Set $\mathcal{A}_m = \left[a_{m_{(q)}-1} + 1 : a_{m_{(q)}}\right]$ for all $m \in [L]$, where $m_{(q)} = ((m+d-1) \bmod J)+1$.

Then, the entropy $H\left(U_{[J]}\right)$ can be upper bounded as follows.

$$\begin{aligned}
H\left(U_{[J]}\right) &= \sum_{m=1}^{J} H(U_m|U_{[m-1]}) \\
&\overset{(a)}{=} H(U_1) + \sum_{m=2}^{J} H(U_m|U_{m-1}) \\
&\overset{(b)}{=} H(U_1) + \sum_{m=2}^{J} \sum_{j=0}^{\theta-1} \mathbb{P}\left(U_{m-1} = j\right) H\left(U_m \middle| U_{m-1} = j\right) \\
&= H\left(\sum_{i \in \mathcal{A}_1} \mathbb{1}\{S_i = \ell\}\right) + \sum_{m=2}^{J} \sum_{j=0}^{\theta-1} \mathbb{P}\left(U_{m-1} = j\right) H\left(\sum_{i \in \mathcal{A}_m} \mathbb{1}\{S_i = \ell\}\right),
\end{aligned} \tag{7.19}$$

where (a) follows since $U_1 \multimap \cdots \multimap U_J$ form a Markov chain and (b) follows since U_m conditioned on $\{U_{m-1} \geq \theta\}$ is deterministic.

Then, Lemma 7.3, (7.17), and (7.18) imply that if $m_{(q)} \in [J-1]$,

$$H\left(\sum_{i \in \mathcal{A}_m} X_i\right) < \frac{1}{2} \log\left(2\pi e\left(\theta + 1 + \frac{1}{12}\right)\right), \tag{7.20}$$

and if $m_{(q)} = J$,

$$\begin{aligned}
H\left(\sum_{i \in \mathcal{A}_m} X_i\right) &< \frac{1}{2} \log\left(2\pi e\left(2\theta + \frac{1}{12}\right)\right) \\
&< \frac{1}{2} + \frac{1}{2} \log\left(2\pi e\left(\theta + 1 + \frac{1}{12}\right)\right).
\end{aligned} \tag{7.21}$$

Hence, (7.19) to (7.21) imply that

$$H\left(U_{[J]}\right) < \frac{1}{2} + \frac{\log\left(2\pi e\left(\theta+\frac{13}{12}\right)\right)}{2}\left(1+\sum_{m=2}^{J}\sum_{j=0}^{\theta-1}\mathbb{P}\left(U_{m-1}=j\right)\right)$$
$$< \frac{1}{2} + \frac{\log\left(2\pi e\left(\theta+\frac{13}{12}\right)\right)}{2}\left(4+\sum_{m=5}^{J}\sum_{j=0}^{\theta-1}\mathbb{P}\left(U_{m-1}=j\right)\right). \quad (7.22)$$

Now we show that the double summation in (7.22) can be upper bounded by a constant independent of J and θ. Denote $\mathcal{S}_0=\emptyset$ and $\mathcal{S}_m=\bigcup_{t=1}^{m}\mathcal{A}_t$ for all $m\in[J]$. For $j\in[1:|\mathcal{S}_{m-1}|]$, $\mathbb{P}\left(U_{m-1}=j\right)=\mathbb{P}\left(Y=j\right)$ where $Y\sim$ Poisson Binomial$(\beta_{\mathcal{S}_{m-1}})$. Denote by $\mathcal{F}_m$ the set of all subsets of $\mathcal{S}_m$ with j elements and let $\Omega_m^*\in\mathcal{F}_m$ be the set of the j indices with the largest values of β_i. Then, we have

$$\begin{aligned}
&\mathbb{P}\left(U_{m-1}=j\right)\\
&=\sum_{\Omega\in\mathcal{F}_m}\prod_{i\in\Omega}\beta_i\prod_{t\in\mathcal{S}_{m-1}\setminus\Omega}(1-\beta_t)\\
&\le\prod_{t\in\mathcal{S}_{m-1}\setminus\Omega_m^*}(1-\beta_t)\sum_{\Omega\in\mathcal{F}_m}\prod_{i\in\Omega}\beta_i\\
&\overset{(a)}{\le}\left(1-\frac{1}{|\mathcal{S}_{m-1}|-j}\sum_{t\in\mathcal{S}_{m-1}\setminus\Omega_m^*}\beta_t\right)^{|\mathcal{S}_{m-1}|-j}\sum_{\Omega\in\mathcal{F}_m}\prod_{i\in\Omega}\beta_i\\
&\overset{(b)}{\le}\left(1-\frac{(m-1)\theta-j}{|\mathcal{S}_{m-1}|-j}\right)^{|\mathcal{S}_{m-1}|-j}\sum_{\Omega\in\mathcal{F}_m}\prod_{i\in\Omega}\beta_i\\
&\overset{(c)}{\le}\left(e^{-1}\right)^{(m-1)\theta-j}\sum_{\Omega\in\mathcal{F}_m}\prod_{i\in\Omega}\beta_i\\
&\le e^{j}e^{-(m-1)\theta}\frac{1}{j!}\left(\sum_{i\in\mathcal{S}_{m-1}}\beta_i\right)^{j}\\
&\overset{(d)}{\le}e^{j}e^{-(m-1)\theta}\frac{(m(\theta+1))^j}{j!},
\end{aligned}$$

where (a) follows since $\prod_i x_i$ is Schur-concave when all $x_i>0$, (b) follows from (7.16) and (7.17), (c) follows since $\left(1-\frac{u}{x}\right)^x\le e^{-u}$ for all $x\ge 1$, and (d) follows from (7.17) and (7.18). Thus,

$$\begin{aligned}
&\sum_{m=5}^{J}\sum_{j=0}^{\theta-1}\mathbb{P}\left(U_{m-1}=j\right)\\
&\le\sum_{m=5}^{J}\sum_{j=0}^{\theta-1}e^{-(m-1)\theta}\frac{(me(\theta+1))^j}{j!}\\
&=e^{-9\theta/4}\sum_{m=0}^{J-4}\sum_{j=0}^{\theta-1}\frac{((m+5)e(\theta+1))^j}{j!}e^{-(m+7/4)\theta}\\
&\overset{(a)}{\le}e^{-5\theta/4}\theta\frac{(\theta+1)^\theta}{\theta!}\sum_{m=0}^{J-4}(m+5)^\theta e^{-(m+7/4)\theta}
\end{aligned}$$

$$\begin{aligned}
&\overset{(b)}{\leq} e^{-5\theta/4}\theta\frac{(\theta+1)^\theta}{\sqrt{2\pi\theta}(\theta/e)^\theta}\sum_{m=0}^{J-4}(m+5)^\theta e^{-(m+7/4)\theta}\\
&= \frac{1}{\sqrt{2\pi}}e^{-\theta/4}\sqrt{\theta}\left(1+\frac{1}{\theta}\right)^\theta\sum_{m=0}^{J-4}\left((m+5)e^{-m-7/4}\right)^\theta\\
&\overset{(c)}{\leq} \sqrt{\frac{e}{\pi}}\sum_{m=0}^{J-4}\left((m+5)e^{-m-7/4}\right)^\theta\\
&\overset{(d)}{\leq} \sqrt{\frac{e}{\pi}}\sum_{m=0}^{\infty}(m+5)e^{-m-7/4} < 1, \qquad (7.23)
\end{aligned}$$

where (a) follows since $\frac{c^j}{j!}$ is an increasing function of j for all $0 \leq j \leq c$, (b) follows from Stirling's formula, (c) follows since $\sqrt{m}e^{-m/4} \leq \sqrt{2/e}$ for all $m \in \mathbb{Z}^+$ and $(1+1/x)^x < e$ for all $x > 0$, and (d) follows since $(m+5)e^{-m-7/4} < 1$ for all $m \geq 0$. Finally, we substitute (7.23) into (7.22) and then the theorem is established after some straightforward simplification. ■

8 Conclusion

In this thesis, we studied efficient information processing for content delivery with caching and for collecting summary statistics in wireless sensor networks, under the paradigm of function computation over networks. The problem of sequential coding for computing was introduced to model content delivery with caching. For the single-user case, we established that caching the most popular components is optimal when the components are independent, caching the coarsest versions is optimal when the the components are nested, and the optimal caching strategy should aim at minimizing the conditional total correlation when the requests are uniformly distributed. Thus, it is fair to say that the most useful cache content is something common or at least popular. For the general multi-user case, we observed two principles "cache $\rightarrow$ update" and "common $\rightarrow$ private" that assist in identifying manageable subproblems for which the optimal rate region admits a single-letter expression. Some progress has been made for distributed computing with successive refinement. In particular, we characterized the optimal rate region of sequential successive refinement and showed that all sources are successively refinable in sum rate as long as full recovery is required in the second stage. For collecting summary statistics in wireless sensor networks, we proposed arithmetic computation coding which achieves a better scaling behavior comparing with naively collecting full data. When interaction among sensors is possible, we showed that type-threshold functions can be computed more efficiently.

Throughout the thesis, we see that the efficiency of computing a function over networks comes from reducing or exploiting redundant information. On the other hand, from the Körner–Marton problem and the problem of sequential coding for computing, we know that completely avoiding redundant information is impossible in distributed function computation. Therefore, characterizing the minimum redundancy is the main challenge in this line of research. Finally, we leave two general research directions that can be extended from this thesis:

- **Outer bounds.** Except the very recent work [42], all known outer bounds for function computation are based on cut-set based arguments, which assume full cooperation among nodes on the same side of the cut. In general there should be a rate penalty due to lack of coordination which cannot be captured by the

cut-set bounds. Thus, any progress for outer bounds is highly important for the field of distributed computing.

- **Efficient evaluation of achievable rate regions.** For practical developments, the achievable rate regions presented in this thesis can serve as benchmarks for indicating the performance of the developed codes. Therefore, efficient evaluation of these single-letter expressions becomes important. Besides, identifying the optimal auxiliary random variables can also provide some insight for code developments.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul. 1948.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 471–480, Jul. 1973.

[3] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 219–221, Mar. 1979.

[4] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, pp. 3498–3516, Oct. 2007.

[5] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[6] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, pp. 903–917, Mar. 2001.

[7] A. D. Wyner, "On source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. IT-21, pp. 294–300, May 1975.

[8] R. Ahlswede and J. Körner, "Source coding with side information and a converse for degraded broadcast channels," *IEEE Trans. Inf. Theory*, vol. IT-21, pp. 629–637, Nov. 1975.

[9] R. M. Gray and A. D. Wyner, "Source coding for a simple network," *Bell Syst. Tech. J.*, vol. 53, pp. 1681–1721, Nov. 1974.

[10] A. Kaspi, "Rate-distortion function when side-information may be present at the decoder," *IEEE Trans. Inf. Theory*, vol. 40, pp. 2031–2034, Nov. 1994.

[11] C. Heegard and T. Berger, "Rate distortion when side information may be absent," *IEEE Trans. Inf. Theory*, vol. IT-31, pp. 727–734, Nov. 1985.

[12] T. M. Cover, "A proof of the data compression theorem of slepian and wolf for ergodic sources," *IEEE Trans. Inf. Theory*, vol. IT-21, pp. 226–228, Mar. 1975.

[13] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inf. Theory*, vol. IT-28, pp. 585–592, Jul. 1982.

[14] I. Csiszár and J. Körner, *Information theory: Coding theorems for discrete memoryless systems*, 2nd ed. Cambridge University Press, 2011.

[15] A. D. Wyner, "A theorem on the entropy of certain binary sequences and applications: Part II," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 772–777, Nov. 1973.

[16] ——, "The common information of two dependent random variables," *IEEE Trans. Inf. Theory*, vol. IT-21, pp. 163–179, Mar. 1975.

[17] A. Sgarro, "Source coding with side information at several decoders," *IEEE Trans. Inf. Theory*, vol. IT-23, pp. 179–182, Mar. 1977.

[18] M. Benammar and A. Zaidi, "Rate-distortion function for a Heegard-Berger problem with two sources and degraded reconstruction sets," in *Proc. IEEE Int. Information Theory Workshop (ITW)*, Jeju Island, Korea, Oct. 2015.

[19] R. Timo, T. J. Oechtering, and M. Wigger, "Source coding problems with conditionally less noisy side information," *IEEE Trans. Inf. Theory*, vol. 60, pp. 5516–5532, Sep. 2014.

[20] E. Perron, S. N. Diggavi, and İ. E. Telatar, "On the role of encoder side-information in source coding for multiple receivers," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seattle, WA, Jul. 2006.

[21] T. Laich and M. Wigger, "Utility of encoder side information for the lossless Kaspi/Heegard-Berger problem," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.

[22] R. Timo, T. Chan, and A. Grant, "Rate distortion with side-information at many decoders," *IEEE Trans. Inf. Theory*, vol. 57, pp. 5240–5257, Aug. 2011.

[23] T. Berger, "Multiterminal source coding," in *The Information Theory Approach to Communications*, ser. CISM Courses and Lectures, G. Longo, Ed. New York: Springer-Verlag, 1978, vol. 229, pp. 171–231.

[24] S.-Y. Tung, "Multiterminal source coding," Ph.D. dissertation, Cornell University, 1978.

[25] R. Ahlswede and T. S. Han, "On source coding with side information via a multiple-access channel and related problems in multi-user information theory," *IEEE Trans. Inf. Theory*, vol. IT-29, pp. 396–412, May 1983.

[26] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, pp. 2856–2867, May 2014.

[27] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," in *arXiv:1308.0178[cs.IT]*, Mar. 2014.

[28] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," in *arXiv:1301.5848[cs.IT]*, Mar. 2014.

[29] U. Niesen and M. A. Maddah-Ali, "Coded caching for delay-sensitive content," in *arXiv:1407.4489[cs.IT]*, Jul. 2014.

[30] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, Australia, Jun. 2014.

[31] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *arXiv:1403.7007[cs.IT]*, Jun. 2014.

[32] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching and coded multicasting: Multiple groupcast index coding," in *Proc. IEEE Global Conf. Signal Info. Processing (GlobalSIP)*, Atlanta, GA, Dec. 2014.

[33] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 355–370, Feb. 2015.

[34] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," in *arXiv:1305.5216[cs.IT]*, Apr. 2014.

[35] J. Hachem, N. Karamchandani, and S. Diggavi, "Coded caching for heterogeneous wireless networks with multi-level access," in *arXiv:1404.6560[cs.IT]*, Apr. 2014.

[36] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," in *arXiv:1502.03124[cs.IT]*, Feb. 2015.

[37] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb. 2015.

[38] H. Viswanathan and T. Berger, "Sequential coding of correlated sources," *IEEE Trans. Inf. Theory*, vol. 46, pp. 236–246, Jan. 2000.

[39] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Sep. 1999.

[40] H. S. Witsenhausen and A. D. Wyner, "A conditional entropy bound for a pair of discrete random variables," *IEEE Trans. Inf. Theory*, vol. 21, pp. 493–501, Sep. 1975.

[41] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM Journal of Research and Development*, vol. 4, pp. 66–82, Jan. 1960.

[42] M. Sefidgaran, A. Gohari, and M. R. Aref, "On Körner–Marton's sum modulo two problem," in *Proc. Iran Workshop Communication and Information Theory (IWCIT)*, Tehran, Iran, May 2015.

[43] V. N. Koshelev, "Hierarchical coding of discrete sources," *Probl. Pered. Inform.*, vol. 16, no. 3, pp. 31–49, 1980.

[44] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, pp. 269–275, Mar. 1991.

[45] B. Rimoldi, "Successive refinement of information: Characterization of the achievable rates," *IEEE Trans. Inf. Theory*, vol. 40, pp. 253–259, Jan. 1994.

[46] Y. Steinberg and N. Merhav, "On successive refinement for the Wyner–Ziv problem," *IEEE Trans. Inf. Theory*, vol. 50, pp. 1636–1654, Aug. 2004.

[47] S. N. Diggavi and C. Tian, "Side-information scalable source coding," *IEEE Trans. Inf. Theory*, vol. 54, pp. 5591–5608, Dec. 2008.

[48] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian–Wolf rate region for arbitrarily correlated sources," in *Proc. IEEE Data Compression Conf. (DCC)*, Snowbird, UT, Mar. 2004.

[49] D. Gündüz, E. Erkip, A. Goldsmith, and H. Poor, "Source and channel coding for correlated sources over multiuser channels," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3927–3944, Sep. 2009.

[50] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, pp. 6463–6486, Oct. 2011.

[51] A. Goldsmith and P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, pp. 1986–1992, Nov. 1997.

[52] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*. Academic Press, 2007.

[53] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 755–764, Apr. 2005.

[54] N. Ma, P. Ishwar, and P. Gupta, "Interactive source coding for function computation in collocated networks," *IEEE Trans. Inf. Theory*, vol. 58, pp. 4289–4305, Jul. 2012.

[55] P. Harremoës, "Binomial and poisson distributions as maximum entropy distributions," *IEEE Trans. Inf. Theory*, vol. 47, pp. 2039 –2041, Jul. 2001.

[56] J. Adell, A. Lekuona, and Y. Yu, "Sharp bounds on the entropy of the poisson law and related quantities," *IEEE Trans. Inf. Theory*, vol. 56, pp. 2299 –2306, May 2010.

[57] N. Ma and P. Ishwar, "Some results on distributed source coding for interactive function computation," *IEEE Trans. Inf. Theory*, vol. 57, pp. 6180–6195, Sep. 2011.

[58] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2006.

[59] L. A. Shepp and J. Olkin, "Entropy of the sum of independent Bernoulli random variables and of the multidimensional distribution," Stanford Univ. , Stanford, CA, Tech. Rep. 131, Jul. 1978.

Curriculum Vitae

Chien-Yi Wang
School of Computer and Communication Sciences
École Polytechnique Fédéral de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
Email: chien-yi.wang@epfl.ch

Education

2011-2015	Dr ès sc., School of Computer and Communication Sciences, École Polytechnique Fédéral de Lausanne (EPFL), Lausanne, Switzerland.
2009-2010	Exchange student at the RWTH Aachen University, Aachen, Germany.
2007-2010	M.Sc., Graduate Institute of Electronics Engineering, National Taiwan University (NTU), Taipei, Taiwan.
2004-2007	B.Sc., Department of Electrical Engineering, National Tsing Hua University (NTHU), Hsinchu, Taiwan.

Publications

Journal Papers and Manuscripts

1. C.-Y. Wang, S. H. Lim, and M. Gastpar, Information-theoretic caching: Sequential coding for computing, submitted to *IEEE Trans. Inf. Theory*, Apr. 2015.

2. C.-Y. Wang, S.-W. Jeon, and M. Gastpar, Interactive computation of type-threshold functions in collocated Gaussian networks, *IEEE Trans. Inf. Theory*, vol. 61, p. 4765-4775, Sep. 2015.

3. S.-W. Jeon, C.-Y. Wang, and M. Gastpar, Computation over Gaussian networks with orthogonal components, *IEEE Trans. Inf. Theory*, vol. 60, p. 7841-7861, Dec. 2014.

4. S.-W. Jeon, C.-Y. Wang, and M. Gastpar, Approximate ergodic capacity of a class of fading two-user two-hop networks, *IEEE Trans. Inf. Theory*, vol. 60, p. 866-880, Feb. 2014.

5. I-W. Lai, C.-Y. Wang, T.-D. Chiueh, G. Ascheid, and H. Meyr, Asymptotic Coded BER Analysis for MIMO BICM-ID with Quantized Extrinsic LLR, *IEEE Trans. Commun.*, vol. 60, pp. 2820-2828, Oct. 2012.

Conference Papers

1. C.-Y. Wang, S. H. Lim, and M. Gastpar, Information-theoretic caching, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Hong Kong, China, Jun. 2015.

2. C.-Y. Wang, and M. Gastpar, On distributed successive refinement with lossless recovery, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Honolulu, HI, USA, Jul. 2014.

3. C.-Y. Wang, S.-W. Jeon, and M. Gastpar, Multi-round computation of type-threshold functions in collocated Gaussian networks, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.

4. S.-W. Jeon, C.-Y. Wang, and M. Gastpar, Computation over Gaussian networks with orthogonal components, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.

5. S.-W. Jeon, C.-Y. Wang, and M. Gastpar, Approximate ergodic capacity of a class of fading 2-user 2-hop networks, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Boston, MA, Jul. 2012.

6. S.-W. Jeon, C.-Y. Wang, and M. Gastpar, Approximate ergodic capacity of a class of fading 222 networks, in *Inf. Theory and Applications Workshop (ITA)*, San Diego, CA, Feb. 2012. (Invited Paper)

7. C.-Y. Wang, I-W. Lai, T.-D. Chiueh, G. Ascheid, and H. Meyr, BER analysis for MIMO BICM-ID assuming finite precision of extrinsic LLR," in *Proc. IEEE Int. Symp. Inf. Theory and its Appl. (ISITA)*, Taichung, Taiwan, 2010.