UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

RAFAEL GREGORIO LUCAS D'OLIVEIRA

Geometry of Communication Channels

Geometria de Canais de Comunicação

Campinas

2017

Rafael Gregorio Lucas D'Oliveira

Geometry of Communication Channels

Geometria de Canais de Comunicação

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática Aplicada.

e

Thesis presented to the Institute of Mathematics, Statistics and Scientific Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Applied Mathematics.

Orientador: Marcelo Firer

Este exemplar corresponde à versão final da Tese defendida pelo aluno Rafael Gregorio Lucas D'Oliveira e orientada pelo Prof. Dr. Marcelo Firer.

Campinas

2017

Agência(s) de fomento e nº(s) de processo(s): CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

L962g

Lucas D'Oliveira, Rafael Gregorio, 1988-
Geometry of communication channels / Rafael Gregorio Lucas D'Oliveira. – Campinas, SP : [s.n.], 2017.

Orientador: Marcelo Firer.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Teoria da informação. 2. Códigos corretores de erros (Teoria da informação). 3. Mergulhos (Matemática). 4. Decodificação por máxima verossimilhança. 5. Decodificação por mínima distância. I. Firer, Marcelo,1961-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Geometria de canais de comunicação
Palavras-chave em inglês:
Information theory
Error-correcting codes (Information theory)
Embeddings (Mathematics)
Maximum likelihood decoding
Minimum distance decoding
Área de concentração: Matemática Aplicada
Titulação: Doutor em Matemática Aplicada
Banca examinadora:
Marcelo Firer [Orientador]
Carlile Campos Lavor
Marcelo Muniz Silva Alves
Mathieu Dutour Sikiric
Olivier Rioul
Data de defesa: 25-05-2017
Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 25 de maio de 2017 e aprovada pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). MARCELO FIRER

Prof(a). Dr(a). CARLILE CAMPOS LAVOR

Prof(a). Dr(a). MARCELO MUNIZ SILVA ALVES

Prof(a). Dr(a). MATHIEU DUTOUR SIKIRIC

Prof(a). Dr(a). OLIVIER RIOUL

As respectivas assinaturas dos membros encontram-se na Ata de defesa

In memory of Michel Deza.

Acknowledgements

To Capes and the Science without Borders Program for their financial support.

To my adviser and dear friend Marcelo Firer for guiding me in this journey.

To my family and friends for their unconditional support.

To Michel for teaching me so much. You are deeply missed.

"Why, anybody can have a brain. That's a very mediocre commodity! Every pusillanimous creature that crawls on the earth or slinks through slimy seas has a brain! Back where I come from, we have universities, seats of great learning where men go to become great thinkers. And when they come out, they think deep thoughts — and with no more brains than you have. But! They have one thing you haven't got! A diploma! Therefore, by virtue of the authority vested in me by the Universita Committeeatum E Pluribus Unum, I hereby confer upon you the honorary degree of Th.D."

(The Wizard of Oz - 1939 film)

Resumo

Abordamos os canais de comunicação a partir de um ponto de vista geométrico. Mostramos que a decodificação por máxima verossimilhança e a decodificação por minima distância são um caso particular de uma forma mais geral de decodificação que pode ser definida para qualquer matriz. Com base nisso, definimos uma equivalência de decodificação e mostramos que ela divide o espaço de matrizes em classes de equivalência que são regiões generalizadas de um arranjo de hiperplanos bem conhecido. Em seguida, definimos uma distância entre essas regiões que mede a probabilidade de um código aleatório ser decodificado incorretamente. Mostramos que esta distância é uma versão ponderada da distância de Kendall tau. Com isso, obtemos uma distância entre canais. Se para um canal existe uma métrica de modo que os decodificadores por máxima verossimilhança e mínima distância coincidem, o canal é metrizavel. Damos caracterizações para um canal ser metrizavel e apresentamos um algoritmo que constrói uma métrica nesse caso. Mostramos também que qualquer métrica, a menos de uma equivalência de decodificação, pode ser mergulhada isometricamente no hipercubo com a métrica de Hamming e, portanto, em termos de decodificação, a métrica de Hamming é universal. Apresentamos um algoritmo que, para qualquer métrica invariante por translação, dá um limite superior na dimensão mínima de tal mergulho. Encontramos também limitantes inferiores e superiores para essa dimensão. No apêndice, apresentamos uma contribuição teórica feita a um trabalho de navegação de mapas.

Palavras-chave: Teoria da Informação, Teoria de Códigos, Mergulhos no Hipercubo, Decodificação por Máxima Verosimilhança, Decodificação por Minima Distância.

Abstract

We approach communication channels from a geometrical viewpoint. We show that maximum likelihood decoding and minimum distance decoding are a particular case of a more general form of decoding which can be defined for any matrix. Based on this we define a decoding equivalence and show that it partitions the space of matrices into equivalence classes which are generalized regions of a well known hyperplane arrangement: the braid arrangement. We then define a distance between these regions which measures the probability of a random code being decoded incorrectly. It is shown that this distance is a weighted variation of the Kendall tau distance. With this, we obtain a distance between channels. If for a channel there exists a metric such that the maximum likelihood and minimum distance decoders coincide, the channel is metrizable. We give characterizations for a channel to be metrizable and present an algorithm which constructs a metric in such a case. We also show that any metric, up to decoding equivalence, can be isometrically embedded into the hypercube with the Hamming metric, and thus, in terms of decoding, the Hamming metric is universal. We then present an algorithm which for any translation invariant metric gives an upper bound on the minimum dimension of such an embedding. We also give lower and upper bounds for this embedding dimension over the set of all such metrics. In the appendix we present the theoretical contribution made to a work on multi-scale navigation.

Keywords: Information Theory, Coding Theory, Hypercube Embeddings, Maximum Likelihood Decoding, Minimum Distance Decoding.

Contents

1 Introduction

In this work we approach communication channels from a geometrical viewpoint. Our interest is not in a specific channel, but rather on the space $Cha_{n\times m}$ of all channels which are studied from the decoding perspective: we consider two channels to be equivalent if, for any given code and any received message (possibly with errors), the message most likely to have been sent is the same. In other words, we define a decoding equivalence where two channels are decoding equivalent if they share the same maximum likelihood decoders. We did not find in the coding and information theory literature any similar approach, so we needed to use tools coming from many different areas of mathematics: geometry of cuts, hyperplane arrangements, intersection patterns, distance embeddings, etc.

As we mentioned, a systematic approach of considering the space of all channels, as far as we know, has not yet been developed. However, it is possible to have some insights for this approach by considering a metric structure matched to a channel. This is a usual procedure in coding theory, with the emblematic use of the Hamming metric when considering a binary symmetric memoryless channel.

Considering this relation between the probabilistic model of the channel and a possible metric approach, we make a similar construction for the space of distances, Dis_n, considering an equivalence determined by minimum distance decoding. In this case we get close to a subject that is well studied in mathematics, a subject called by Michel Deza as "geometry of cuts and metric", the title of his monograph with M. Laurent. However, in this common setting, the equivalence between distances used is the scalar equivalence, where two distances differ only by a multiplicative constant. We contrast this notion with our decoding equivalence which is a novel concept in mathematics, not explored in the literature.

We then show that maximum likelihood decoding and minimum distance decoding are a particular case of a more general form of decoding. We consider the space of all positive matrices, $\mathbb{R}^{n\times m}$, and define a decoding equivalence such that both maximum likelihood and minimum distance decoders are particular cases.

Our first goal is to characterize the decoding equivalence classes of channels. Fortunately, those can be described in a simple way: the decoding equivalence partitions $\mathbb{R}^{n\times m}$ into disjoint cones which we call decoding cones. The way that this happens is intimately related to a field of study called Hyperplane Arrangements. The decoding cones are related to the regions of a well studied hyperplane arrangement called the braid arrangement.

The next step is to establish an appropriate distance *between* decoding cones. The usual distance used in this context is a Cayley distance determined by a simple set of roots (for group theorists) also known as the Kendall tau distance. However, this distance is inappropriate for our purpose. Indeed, each decoding cone determines a decision criterion for decoding a given code. To say that two cones are different implies they determine different criteria for *some* code, but not for every code. The Kendall tau distance does not measure the probability of a random code to be *misdecoded*. We define a weighted variation of the Kendall tau distance which achieves this goal.

Another question arises in the case $n = m$, since in this case both $Cha_{n\times n}$ and Dis_n coexist as subsets of $\mathbb{R}^{n\times n}$. In this situation, it is meaningful to ask whether a channel is is metrizable, i.e. there exists a metric such that maximum likelihood decoding and minimum distance decoding are the same. We give sufficient and necessary conditions to a positive answer of this question. This condition can be seen as a description of a specific cone which intercepts Dis_n: this specific cone is the cone of metrizable channels. We also give an algorithm that either determines a distance matched to the channel or decide that such a distance does not exist.

In the context of Deza's theory of distance embedding, a starting crucial question is to find a kind of *universal space*, a distance space where any other can be isometrically embedded. In our context, considering the metrizable channels, we show that in terms of decoding, the Hamming distance is universal, i.e. that every metric is, up to decoding equivalence, is isometrically embeddable into the Hamming cube. Moreover, when a distance is compatible with a linear structure, this embedding is linear. To make such an achievement, looking at every possible instance as a subcase of the Hamming instance, there is a price to be paid: we need to increase the dimension of the space. Using tools developed in the area of intersection patterns, we give bounds for the minimum dimension of such an embedding.

Finally, this thesis has a final chapter which presents the theoretical contribution made to a work on multi-scale navigation. Since the work was recognized by specialists as a relevant contribution it deserves to be a part of this thesis. However, since the subject is only marginally related to the main subject of this thesis, it is placed in a "marginal" place: the appendix.

We remark that most of the content in this thesis was published or accepted for publication as journal papers ([12, 13]) and conference papers ([11]). We note that all the propositions stated as so in this thesis are original, except for Proposition 1.

This work is organized in the following manner:

Chapter 2 introduces the basic concepts used throughout our work.

Chapter 3 introduces the key notion of our work: decoding equivalence. It is

here that we present the geometrical frame used throughout our work.

In Chapter 4 a geometrical and a graph theoretical characterization of channel metrization are given. An algorithm for determining channel metrization is given.

In Chapter 5 we show that any metric, up to a decoding equivalence, can be isometrically embedded into the hypercube with the Hamming metric, and thus, in terms of decoding, the Hamming metric is universal. We also study the dimension of the embedding.

In Chapter 6 we discuss future perspectives for our work, a relevant chapter since the amplitude of the subject arises many questions that are not answered.

Finally, Appendix A is devoted to the contribution made to multi-scale navigation.

2 Basic Concepts and Notation

In this chapter we present the basic notions which we will use throughout our work. These concepts are well known in their respective fields. We tried to make this work as self contained as possible, but just in case, we give basic bibliographic references to the contents of each section. Many of those concepts will be defined again the first time they are needed, so this chapter functions also as an “extended index of notations".

In Section 2.1 we define the Iverson Bracket. These were first introduced by Iverson in [21] and later popularized by Knuth in [25].

Section 2.2 is about the notion of order. We will see that a channel induces a certain ordering among messages which are transmitted through it. The book [35] is a basic introductory reference.

Section 2.3 is about discrete geometry. Our geometrical approach will use many concepts from this area. Hyperplane arrangements will be particularly important. Details can be found in [18] and [38].

Section 2.4 is about distances. Throughout coding theory it is more common to consider the less general notion of a metric. We will see however that if one is interested in decoding, the triangle inequality has no essential relevance and therefore, nothing is lost (but actually gained) by considering more general distances. Many variations on the subject can be found in [8].

Section 2.5 is about channels. They are one of the main concepts throughout coding theory appearing in Shannon‘s celebrated model of communication. A suggested systematic introduction is the book [5].

Section 2.6 is about decoders. They also appear in Shannon‘s model of communication and in a sense are our main object of study. See [32] in case more details are needed.

2.1 The Iverson Bracket: $[P]$

The *Iverson Bracket* converts any logical proposition P into 1 if it is true and 0 if it is false, i.e. $[P] = \begin{cases} 1 & \text{if P is true} \\ 0 & \text{otherwise} \end{cases}$.

2.2 Orders

A *preorder* over a set X is a triple $(X, \prec, \simeq)$, where $\prec$ and $\simeq$ are binary relations on X satisfying

1. for all $x \in X$, it is not the case that $x \prec x$,
2. for all x, y, z, $x \prec y$ and $y \prec z$ implies that $x \prec z$,
3. $\simeq$ is an equivalence relation,
4. for all $x, y \in X$, at most one of $x \prec y$, $y \prec x$ or $x \simeq y$ holds.

We denote the set of all preorders over n elements by Pro_n.

If either $x \prec y$, $y \prec x$, or $x \simeq y$, we say that x and y are *comparable*.

A *preorder* can be interpreted as a mixed graph (a graph with directed and undirected edges), such that there is a directed edge from x to y if $x \prec y$ and an undirected one if $x \simeq y$. We will often identify the preorder with its graph.

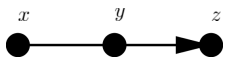

Figure 1 – Mixed graph for $x \simeq y \prec z$

A *weak order* is a preorder in which every element is comparable. We denote the set of all weak orders over n elements by W_n.

The number of weak orders are known as the *Ordered Bell numbers* $|W_n| = \sum_{k=0}^{n} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$, which exceed the factorials by an exponential factor.

A linear extension of a preorder $(X, \prec, \simeq)$ is a weak order on the same set $(X, \prec', \simeq')$ which preserves the preorder, i.e. if $x \prec y$ ($x \simeq y$ respectively) holds then so does $x \prec' y$ ($x \simeq' y$ respectively).

Determining the number of linear extensions for general preorders is $\#P$-complete.

2.3 Discrete Geometry

A set $A \subseteq \mathbb{R}^n$ is *convex* if it contains the segment joining any two of its points, i.e. $\alpha x + (1 - \alpha) y \in A$ for every $x, y \in A$ and $0 \leqslant \alpha \leqslant 1$.

A *hyperplane* is a set $H \subseteq \mathbb{R}^n$ of the form $H = \{x \in \mathbb{R}^n : \alpha \cdot x = a\}$ where $0 \neq \alpha \in \mathbb{R}^n$, $a \in \mathbb{R}$ and $\alpha \cdot x := \sum_{i=1}^{n} \alpha_i x_i$ is the usual dot product.

A *hyperplane arrangement*, $\mathcal{A}$, is a set of hyperplanes. A region of an arrangement is a connected component of the complement of the hyperplanes, $X = \mathbb{R}^n - \bigcup_{H \in \mathcal{A}} H$. The set of regions is denoted by $\mathcal{R}(\mathcal{A})$ and $r(\mathcal{A}) := \#\mathcal{R}(\mathcal{A})$.

Each hyperplane divides $\mathbb{R}^n$ into two subsets known as *half-spaces*. The two half spaces corresponding to $H = \{x \in \mathbb{R}^n : \alpha \cdot x = a\}$ are $\{x \in \mathbb{R}^n : \alpha \cdot x \leqslant a\}$ and $\{x \in \mathbb{R}^n : \alpha \cdot x \geqslant a\}$.

A *convex polytope* is the intersection of a finite set of half-spaces.

A set $C \subseteq \mathbb{R}^n$ is a *convex cone* if $\alpha x + \beta y \in C$ for every $x, y \in C$ and $\alpha, \beta \geqslant 0$.

The *conical combination* of a finite number of vectors, called *generators of the cone*, $x_1, x_2, \ldots, x_k \in \mathbb{R}^n$ is the set $coni(x_1, x_2, \ldots, x_k) = \{\sum_{i=1}^{n} \alpha_i x_i : \alpha_i \geqslant 0\}$. This set is a convex cone and a convex cone which can be expressed in such a way is said to be *finitely generated.*

The *extreme rays* of the cone C are the one-dimensional faces and form a minimal set of generators of C.

We are particularly interested in the *braid arrangement,* $\mathcal{B}_n$, which consists of the $\binom{n}{2}$ hyperplanes: $x_i - x_j = 0$ for $1 \leqslant i < j \leqslant n$. Specifying to which side of the hyperplane a point $a \in \mathbb{R}^n$ belongs to is equivalent to determining whether $a_i < a_j$ or $a_j < a_i$. Doing so for every hyperplane is equivalent to imposing a linear order on the a_i. So to each permutation $\sigma \in S_n$ there corresponds a region $R_\sigma \in \mathcal{R}(\mathcal{B}_n)$ given by $R_\sigma = \{x \in \mathbb{R}^n : a_{\sigma(1)} < a_{\sigma(2)} < \ldots < a_{\sigma(n)}\}$. Thus, $r(\mathcal{B}_n) = n!$.

2.4 Distances

A *distance* on a set X is a function $d : X \times X \to \mathbb{R}$ which satisfies

1. $d(x, y) \geqslant 0$ (non-negativity)
2. $d(x, y) = d(y, x)$ (symmetry)
3. $d(x, x) = 0$ (reflexivity)

If the distance also satisfies property 4 it is called a *semimetric* and if in addition it satisfies property 5 it is called a *metric*. We denote the set of all distances over a set X with n elements by $Dis_n(X)$. Since the set itself is immaterial to any theoretical purpose, we do not specifiy it and denote $Dis_n = Dis_n(X)$.

4. $d(x, y) = 0$ if and only if $x = y$ (identity of indiscernibles)

5. $d(x,z) \leqslant d(x,y) + d(x,z)$ (triangle inequality)

If the set X is an abelian group such that $d(x+z, y+z) = d(x,y)$ for every $x, y, z \in X$, d is *translation invariant*. In this case the distance function is completely determined by a weight function $\omega : X \to \mathbb{R}$ given by $\omega(x) = d(x,0)$.

The Hamming distance on the set $\mathbb{F}_2^n$ is the translation invariant metric defined as $d_H(x,y) = \#\{x_i \neq y_i : i \in [n]\}$. All the main distances used in coding theory are distance translation invariant: Lee metric, poset metrics, combinatorial metrics (see [16] and [8, Chapter 16] for details on some of those distances).

If the set $X = \{x_1, x_2, \ldots, x_n\}$ is finite with n elements then we can identify the distance function with the matrix d such that $d_{ij} = d(x_i, x_j)$. In this sense, a distance is a non-negative symmetric matrix with zero diagonal.

Since we are mainly interested in semimetrics, we denote the set of all semimetrics over n elements by Sem_n. We denote by $\mathbb{R}^{n\times m}$ the space of $n \times m$ matrices with entries in $\mathbb{R}$. Then, both $Dis_n, Sem_n \subseteq \mathbb{R}^{n\times n}$ are convex cones.

2.5 Channels

In [37] Shannon introduced his famous model of communication. It consists in the following

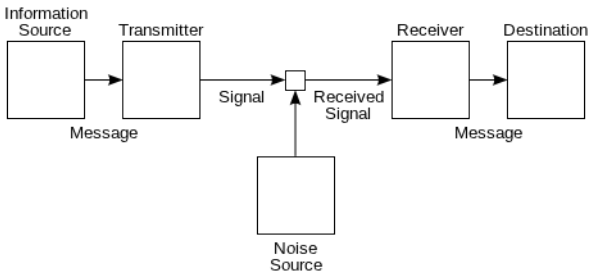

Figure 2 – Shannon‘s original model for a general communication system.

1. An information source which produces messages.
2. A transmitter which transforms the message into some suitable signal for transmission.
3. A channel, the medium used to transmit the signal.
4. A receiver which must reconstruct the message from the signal received.
5. The destination, for whom the message is intended.

We will consider only the case of finite, hence discrete, channels. Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of input messages which the transmitter can send and $Y = \{y_1, y_2, \ldots, y_m\}$, the set of output messages which the receiver can receive. A *channel* is an $n \times m$ probability matrix P such that $P_{ij} = Pr(y_j \text{ is received} \,|\, x_i \text{ is sent})$. We denote the set of all channels with n inputs and m outputs by $Cha_{n\times m}$.

As with distances, the set of channels is also a subset of $\mathbb{R}^{n\times m}$. In fact, $Cha_{n\times m} \subset \mathbb{R}^{n\times m}$ is a convex polytope. This follows from the fact that a matrix is a channel if and only if it satisfies the following inequalities: $0 \leqslant P_{ij} \leqslant 1$ and $\sum_{j=1}^{m} P_{ij} = 1$.

2.6 Decoders

A *code* is a subset $C \subseteq X$ of the set of all possible inputs which the transmitter can send. The idea is that by restricting the set of *codewords* (elements of the code) which can be sent, one can increase the chance of the receiver to interpret correctly what the original message was. As is usual, we assume that the probability distribution for sent messages is uniform, i.e. $Pr(c \text{ is sent}) = \dfrac{1}{|C|}$.

In this work, a *decoder* for a code C (for every decoder we assume a code) is a function $f : Y \to C$ which for every output message $y \in Y$ chooses a codeword $c \in C$. When the receiver receives the message $y \in Y$ he interprets this as $f(y) \in C$ being sent. When $f(y)$ was indeed sent we say that the decoding was done correctly. A decoder can be represented by a matrix D such that $D_{ij} = [f(y_j) = x_i]$. Given a channel and a code, the probability that sending a codeword and after decoding to get the original codeword, this is what we call the *probability of correect decoding* of the code.

This probability can be expressed in terms of the *Frobenius inner product* $\langle P, D\rangle_F := \sum_{i,j} P_{ij}D_{ij}$. Since we could not find a reference for this, we present here a short proof.

Proposition 1. *Let X be the set of input messages, Y the set of output messages and P be the channel. If $C \subseteq X$ is a code and D a decoder. Then, the probability that the receiver will decode correctly is given by*

$$Pr(\textit{correct decoding}) = \frac{\langle P, D\rangle_F}{|C|}.$$

Proof.

$$\begin{aligned} Pr(\text{correct decoding}) &= \sum_{y \in Y} Pr(\text{correct decoding} \,|\, y \text{ is received}) Pr(y \text{ is received}) \\ &= \sum_{y \in Y} Pr(f(y) \text{ is sent} \,|\, y \text{ is received}) Pr(y \text{ is received}) \\ &= \sum_{y \in Y} \frac{Pr(y \text{ received} \,|\, f(y) \text{ sent}) Pr(f(y) \text{ sent})}{Pr(y \text{ received})} Pr(y \text{ received}) \\ &= \frac{1}{|C|} \sum_{y \in Y} Pr(y \text{ received} \,|\, f(y) \text{ is sent}) \end{aligned}$$

□

We are generally interested in maximizing the probability of correct decoding. It follows from Proposition 1 that this can be achieved by selecting a decoder such that for each $y \in Y$, $f(y)$ is chosen as to maximize $Pr(y \text{ is received} \,|\, f(y) \text{ is sent})$. These are known as maximum likelihood decoders. We denote the set of *maximum likelihood decoders* for a channel P with a code C by $D\hat{e}c_C(P)$.

Maximum likelihood decoding is in general a hard problem. Some kind of structure on the channel can often times make it easier and help in the construction of better codes. One such structure is when a distance can be defined between messages such that messages closer to a codeword are more likely to be originated from them.

Suppose that $X = Y$ and that $d : X \times X \mapsto \mathbb{R}$ is a distance. A *minimum distance decoder* is a decoder $f : X \to C$ such that for each $x \in X$, $f(x)$ is chosen as to minimize $d(x, f(x))$. We denote the set of minimum distance decoders for a distance d with a code C by $D\hat{e}c_C(d)$.

When a channel is such that there is a minimum distance decoder which is also a maximum likelihood decoder, we say that the channel is *metrizable* and that the channel and the distance are *matched* to each other.

As an example, the binary symmetric channel, the most commonly studied discrete channel is matched to the Hamming metric. This gives the Hamming metric a prominent status among other distances.

3 Geometry of Communication Channels

The notion of a space of channels, $Cha_{n\times m}$, has not, up to the author's knowledge, been studied before. On the other hand, the notion of a space of distances, Dis_n, is common in mathematics [1]. In the case of distances over finite sets, the regular notion of equivalence is equivalence by scale, i.e. two distances d and d' over a space X are *scalar-equivalent* if there is a constant $\lambda > 0$ such that $d(x, y) = \lambda d'(x, y)$ for all $x, y \in X$. One of our main contributions is a weaker form of equivalence, called decoding equivalence, which arises naturally in the context of coding theory.

We extend the definition of decoding to general matrices so that distances and channels are seen as particular cases. With this, our main object of study becomes the space of matrices under the decoding equivalence.

In Section 3.1 we introduce the key notion of our work: the decoding equivalence.

In Sections 3.2 and 3.3 we show that the decoding equivalence partitions $\mathbb{R}^{n\times m}$ into generalized regions of the braid arrangement, $\mathcal{B}_n$.

In Sections 3.4 and 3.5 we define the decoding distance between between decoding cones and show that it is a weighted version of the Kendall tau distance. We then extend this to a quasidistance between channels.

3.1 Decoding Equivalence

Consider the space of all distances over n elements Dis_n. As seen in Section 2.6, given a distance d and a code C, we denote by $D\hat{e}c_C(d)$ the set of all minimum distance (relatively to d) decoders of the code C. Since we are interested in decoding, it is natural to consider the following equivalence relation.

Definition 1. *Two distances $d, d' \in Dis_n$ are* decoding equivalent*, denoted by $d \sim d'$, if $Dec_C(d) = Dec_C(d')$ for every code $C \subset X$.*

In fact, $d \sim d'$ if and only if they preserve the *weak ordering* (ordering allowing ties) of the distances from a fixed point.

Theorem 1. *Let $d, d' \in Dis_n$. Then, $d \sim d'$ if and only if for every $x, y, z \in X$, it holds that $d(x, z) < d(y, z)$ if and only if $d'(x, z) < d'(y, z)$.*

Proof. Let $d \sim d'$. Take $z \in X$ and suppose that there are $x, y \in X$ such that $d(x, z) <$

$d(y,z)$. Consider the code $C = \{x, y\}$. Then,

$$\arg\min\{d(c,z); c \in C\} = \{x\}.$$

Since $d \sim d'$, it follows that

$$\arg\min\{d'(c,z); c \in C\} = \{x\},$$

and therefore, $d'(x,z) < d'(y,z)$.

In the case that that $d'(x,z) < d'(y,z)$, the analogous arguments would show that $d(x,z) < d(y,z)$. Hence,

$$d(x,z) < d(y,z) \Leftrightarrow d'(x,z) < d'(y,z).$$

Now suppose that for every $x, y, z \in X$

$$d(x,z) < d(y,z) \Leftrightarrow d'(x,z) < d'(y,z).$$

Let C be a code and fix $z \in X$. Suppose that

$$\arg\min\{d(c,z) : c \in C\} \neq \arg\min\{d'(c,z) : c \in C\}.$$

Without loss of generality, we can suppose that there exists a $y \in C$ such that

$$y \in \arg\min\{d(c,z) : c \in C\},\ \ y \notin \arg\min\{d'(c,z) : c \in C\},\ \text{and} |C| \geqslant 2.$$

Let $x \in \arg\min\{d'(c,z) : c \in C\}$. Then, it holds that $d'(x,z) < d'(y,z)$. But this implies, by equivalence, that $d(x,z) < d(y,z)$, and therefore, since $x \in C$,

$$y \notin \arg\min\{d(c,z) : c \in C\},$$

a contradiction. Finalizing the proof. □

Theorem 1 implies directly that two equivalent distances will have the same set of balls, not necessarily for the same or proportional radii. We call $B_d(x,r) = \{y \in X : d(y,x) \leqslant r\}$ the *d-ball centered at x and radius r*.

Corollary 1. *Let $d, d' \in Dis_n$. Then, $d \sim d'$ if and only if for every $x_0 \in X$ and $r_1 \in \mathbb{R}$ there exists an $r_2 = r_2(x_0, r_1) \in \mathbb{R}$ such that $B_{d_1}(x_0, r_1) = B_{d_2}(x_0, r_2)$.*

We now give an example which contrasts the decoding equivalence with the usually studied scalar equivalence ($d \overset{\text{scalar}}{\sim} d'$ if there exists $\lambda \in \mathbb{R}$ such that $d = \lambda d$).

Example 1. *Consider the space of all triangles.*

The scalar equivalence partitions this space into an infinite number of equivalence classes: two triangles are scalar-equivalent if and only if they are similar.

The decoding equivalence partitions this space into 4 equivalence classes:

- $a = b = c$ *(equilateral triangles)*
- $a < b = c$ *(isosceles triangles with vertex angle* $< 60°$*)*
- $a = b < c$ *(isosceles triangles with vertex angle* $> 60°$*)*
- $a < b < c$ *(scalene triangles)*

where a, b, c *denote the length of the sides of the triangle.*

Note that scalar equivalence implies decoding equivalence (if $d \overset{\text{scalar}}{\sim} d'$ then $d \sim d'$).

The distances used in coding theory are usually metrics. However, every semimetric is decoding equivalent to some metric as shown by the following distance transform (a particular case of the "squeezing" argument in [15]).

Example 2. *Let* $d \in Sem_n$ *be a semimetric. Consider the distance transform*

$$d'(x, y) = 1 + \frac{d(x, y)}{\max_{u,v} d(u, v)}$$

for $x \neq y$ *and zero otherwise. Then,* $d \sim d'$ *and* d' *is a metric.*

In terms of the matrix representation two distances are equivalent if the weak orderings of the elements of each column are the same. Since this is also true for channels under maximum likelihood decoding, we will define the following concept for general matrices.

Definition 2. *Given a matrix* $M \in \mathbb{R}_{\geqslant 0}^{n \times m}$, its decreasing column weak ordered matrix *is the matrix* $O^- M$ *such that* $(O^- M)_{ij} = k$ *if* M_{ij} *is the* k*th largest element (allowing ties) of the* j*th column.*

Similarly, $(O^+ M)_{ij} = k$ *if it is the* k*th smallest element of the* j*th column.*

Example 3. *If* $M = \begin{pmatrix} 9 & 2 & 1 \\ 9 & 7 & 0 \\ 8 & 6 & 8 \end{pmatrix}$, *then*

$$O^- M = \begin{pmatrix} 1 & 3 & 2 \\ 1 & 1 & 3 \\ 2 & 2 & 1 \end{pmatrix} \quad \text{and} \quad O^+ M = \begin{pmatrix} 2 & 1 & 2 \\ 2 & 3 & 1 \\ 1 & 2 & 3 \end{pmatrix}$$

Note that $O^- M = O^- N$ if and only if $O^+ M = O^+ N$.

Corollary 2. *Let* d_1 *and* d_2 *be two distances over* $[n]$. *Then,* $d_1 \sim d_2$ *if and only if* $O^- d_1 = O^- d_2$ *or equivalently, if* $O^+ d_1 = O^+ d_2$.

Analogously to the case for distances, we have the following equivalence relation between channels.

Definition 3. *Two channels, M and N, are called* decoding equivalent*, $M \sim N$, if for any code $C \subset X$, they define the same maximum likelihood decoder.*

The process of maximum likelihood or minimum distance decoding work essentially in the same way. The only difference is that with the MLD we are searching for the largest entry in a column (restricted to the rows corresponding to codewords) while with the MDD we are looking for the smallest entry. In both cases, only the weak ordering of the elements in the columns is important. Let us state it in a precise way.

We first show that this is also the case for channels.

Proposition 2. *Let M and N be two channels over $[n]$. Then, $M \sim N$ if and only if for every $i, k, j \in [n]$*

$$M_{i,j} < M_{k,j} \Leftrightarrow N_{i,j} < N_{k,j}.$$

Proof. The proof is nearly identical to that of Theorem 1, just substituting the distance by the probability matrix and exchanging the minimality condition (related to the distance) by the maximality condition (determined by the probability). □

Corollary 3. *Let M and N be two channels over $[n]$. Then, $M \sim N$ if and only if $O^-M = O^-N$ or equivalently $O^+M = O^+N$.*

We can therefore define decoding equivalence for any two matrices in $\mathbb{R}_{\geqslant 0}^{n\times m}$.

Definition 4. *Two matrices $M, N \in \mathbb{R}_{\geqslant 0}^{n\times m}$ are decoding equivalent, denoted by $M \sim N$, if $O^-N = O^-M$*

This equivalence when restricted to channels or distances coincides with their corresponding decoding equivalence.

3.2 Decoding Equivalence in $\mathbb{R}_{\geqslant 0}^n$

We first define the *Order* function.

Definition 5. *The Order function, $Order : \mathbb{R}_{\geqslant 0}^n \to W_n$, takes a vector $x \in \mathbb{R}_{\geqslant 0}^n$ to the weak ordering of its coordinates.*

So, for example, $Order(\sqrt{2}, \frac{-1}{2}, \sqrt{2}) = Order(2, 1, 2) = (2 \prec 1 \simeq 3)$.

Proposition 3. *Two vectors $x, y \in \mathbb{R}_{>0}^n$ are decoding equivalent if and only if $Order(x) = Order(y)$.*

Proof. This follows because $Order(x) = Order(y)$ if and only if $O^- x = O^- y$. □

The *fibers* of the $Order$ function, i.e., the inverse images $Order^{-1}(y)$, partition $\mathbb{R}^n$ into the decoding equivalence class.

Definition 6. *The cone function is given by* $Cone : \mathbb{R}^n_{\geqslant 0} \to 2^{\mathbb{R}^n_{\geqslant 0}}$ *such that* $Cone(x) = (Order)^{-1} \circ Order$. *We call* $Cone(x)$ *the decoding cone of* x.

We need to generalize the definition of the region of a hyperplane arrangement.

Definition 7. *A generalized region of a hyperplane arrangement* $\mathcal{A}$ *is a connected component of* $\bigcap_{H \in \mathcal{A}_1} H - \bigcup_{H \in \mathcal{A}_2} H$, *where* $\mathcal{A}_1, \mathcal{A}_2$ *is a disjoint partition of* $\mathcal{A}$. *We denote the sets of generalized regions by* $\mathcal{GR}(\mathcal{A})$ *and* $gr(\mathcal{A}) = \#\mathcal{GR}(\mathcal{A})$.

As stated in Section 2.3, the braid arrangement consists of the $\binom{n}{2}$ hyperplanes: $H_{ij} = \{x \in \mathbb{R}^n : x_i = x_j\}$ for $1 \leqslant i < j \leqslant n$. The next theorem shows that the decoding equivalence partitions $\mathbb{R}^n$ into generalized regions of the braid arrangement.

Theorem 2. *Let* $x, y \in \mathbb{R}^n$. *Then,* x *is decoding equivalent to* y *if and only if* $x, y \in R$ *for some* $R \in \mathcal{GR}(\mathcal{B}_n)$, *where* $\mathcal{B}_n$ *is the braid arrangement.*

Proof. Specifying to which generalized region $R_x \in \mathcal{GR}(\mathcal{B}_n)$ a point $x \in \mathbb{R}^n$ belongs to is equivalent to determining whether $x_i < x_j$, $x_i = x_j$ or $x_i > x_j$ for every $1 \leqslant i < j \leqslant n$. This is equivalent to imposing a weak order on the coordinates of x. But this implies that $y \in R_x$ if and only if $Order(y) = Order(x)$. The result then follows from Proposition 3. □

In other words, if $R \in \mathcal{GR}(\mathcal{B}_n)$ then $x \in R$ if and only if $R = Cone(x)$, i.e. the decoding cones are the generalized regions of the braid arrangement.

Proposition 4. *Let* $x \in \mathbb{R}^n_{\geqslant 0}$. *The extreme rays of* $Cone(x)$ *are given by* $\{g^1, g^2, \ldots, g^n\}$ *where* $g^i_j = [x_i \leqslant x_j]$, *for* $i, j \leqslant n$.

Proof. By Theorem 2 the extreme rays of $Cone(x)$ are the same as those of the generalized regions of the braid arrangement. □

Example 4. *If* $d = (1, 2, 3, 2)$, *then* $Cone(d)$ *is generated by*

$$g^1 = (1, 1, 1, 1), \ g^2 = (0, 1, 1, 1) \text{ and } g^3 = (0, 0, 1, 0).$$

Note that the all one vector, which we denote by $\overrightarrow{\mathbf{1}}$, is always an extreme ray.

The dimension of the cone depends on the number of equalities in the ordering.

Proposition 5. *Let $x \in \mathbb{R}^n_{\geqslant 0}$ and e be the number of undirected edges in the mixed graph $Order(x)$. Then, the dimension of $Cone(x)$ is $n - e$.*

Proof. By Theorem 2, $Cone(x) \in \mathcal{GR}(\mathcal{B}_n)$. By Definition 7, $Cone(x)$ is a connected component of $\bigcap_{H \in \mathcal{A}_1} H - \bigcup_{H \in \mathcal{A}_2} H$, for some partition $\mathcal{A}_1, \mathcal{A}_2$ of $\mathcal{B}_n$.

Thus, $Cone(x)$ has the same dimension as $\bigcap_{H \in \mathcal{A}_1} H$. But this is the intersection of $\#\mathcal{A}_1$ hyperplanes. Thus, the dimension of $Cone(x)$ is equal to $n - \#\mathcal{A}_1$. Since $\mathcal{A}_1$ is a partition of the braid arrangement, each hyperplane corresponds to some equality in the coordinates of x. Thus, $\#\mathcal{A}_1 = e$, from where the result follows.

□

We denote by $\left\{ {n \atop k} \right\}$ the number of ways to partition a set of size n into k nonempty subsets. These are called *Stirling numbers* of the second kind.

Proposition 6. *The number of $(n-k)$-dimensional decoding cones in $\mathbb{R}^n$ is $k! \left\{ {n \atop k} \right\}$.*

Proof. In the proof of Theorem 2 we showed that each decoding cone corresponds to a weak order on n. In Proposition 5 we showed that the dimension of $Cone(x)$ is determined by the number of equalities in $Order(x)$. Thus, the number of $(n-k)$-dimensional decoding cones is equal to the number of weak orders with k equalities. The number of ways to set these equalities is $\left\{ {n \atop k} \right\}$ and to order them is $k!$. □

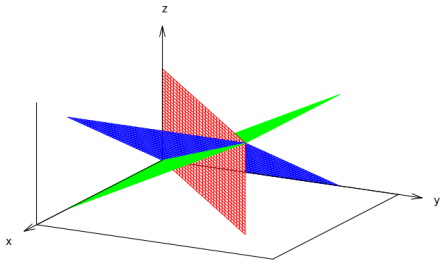

Figure 3 – The partition of $\mathbb{R}^3_{>0}$ by decoding equivalence into 13 cones: six 3-dimensional, six 2-dimensional, and one 1-dimensional (the ray $(\lambda, \lambda, \lambda)$ with $\lambda > 0$).

3.3 Decoding Equivalence in $\mathbb{R}^{n \times m}_{\geqslant 0}$

In the previous section we considered the ordering of one single string. In terms of decoding, it is equivalent to establishing the order of preference for decoding once one given message is received. In this section we wish to deal with all possible received messages

simultaneously, so that we establish our preferences without knowing which message was received. In this sense we extend the results of the previous one from $\mathbb{R}^n$ to $\mathbb{R}^{n\times m}$. We will abuse notation and use the same names.

Definition 8. *The Order function,* $Order : \mathbb{R}^{n\times m}_{\geqslant 0} \to W^m_n$*, is defined as*

$$Order(M) = Order(M[\cdot][1]) \times Order(M[\cdot][2]) \times ... \times Order(M[\cdot][m]),$$

where $Order(M[\cdot][j])$ *is the the order function in Definition 5 applied to the* j*-th column of* M*. The decoding cone of* M *is* $Cone(M) = Order^{-1} \circ Order(M)$*.*

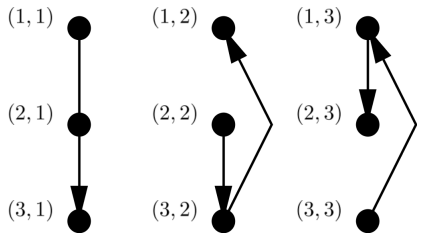

Figure 4 – Order(M), for Example 3.

We get the following two results as direct analogs of the $\mathbb{R}^n$ case.

Theorem 3. *Let* $M, M' \in \mathbb{R}^{n\times m}$*. Then,* M *is decoding equivalent to* M' *if and only if* $M[\cdot][j], M'[\cdot][j] \in R_j$ *for some* $R_j \in \mathcal{GR}(\mathcal{B}_n)$*, where* $\mathcal{B}_n$ *is the braid arrangement.*

Proof. The proof is equivalent to that of Theorem 2 by using Definition 8. □

Proposition 7. *Let* $M \in \mathbb{R}^{n\times m}_{\geqslant 0}$ *and* e *be the number of undirected edges in the mixed graph* $Order(M)$*. Then, the dimension of* $Cone(M)$ *is* $mn - e$*.*

Proof. The proof is equivalent to that of Proposition 5 by using Definition 8. □

A special case of interest is in the case where $n = m$. When this happens, both distances (Sem_n) and channels (Cha_n) are subsets of the same space ($\mathbb{R}^{n\times n}_{\geqslant 0}$). It is in this space where channels can be matched to metrics. In Section 4.1 we will see that a channel $P \in Cha_n$ is metrizable if and only if $Cone(P) \cap Sem_n \neq \varnothing$.

3.4 A Decoding Distance Between Permutations

Having an appropriate model of the transmission channel is not always enough to establish all the necessities in the communication process. Many other questions, such as the complexity of the decoding algorithms, needs to be taken into consideration. For this reason, for example, the Hamming metric is many times used, even when the channel is not the binary symmetric channel or any other to which it is matched.

In this sense, it may be interesting to develop an "approximation theory" for channels. The idea is that we can use an approximative channel (or distance) in place of the original one.

A powerful tool for the development of an approximation theory is to have a distance in the space $Cha_{n\times m}$ which is adequate in some sense. If P is a channel, P_y denotes the column corresponding to y being a received message. We will propose a relevant distance on $Cha_{n\times m}$ which answers the following question:

Let $P, Q \in Cha_{n\times m}$ be two different channels and suppose we know what output $y \in Y$ is received. Choosing a code $C \subseteq X$ from the set of all codes with uniform distribution, what is the probability that $D\hat{e}c_C(P_y) \cap D\hat{e}c_C(Q_y) \neq \varnothing$?

When we say that the distance will answer that question it means that the probability that $D\hat{e}c_C(P_y) \cap D\hat{e}c_C(Q_y) \neq \varnothing$ decreases with the purposed distance.

Since we know what output y is received, only the column corresponding to it matters for decoding. Thus we are dealing with the decoding equivalence in $\mathbb{R}^n$.

We will only consider the cases for which $Cone(P_y)$ and $Cone(Q_y)$ are n-dimensional (and leave the general case for future work). In this case $D\hat{e}c_C(P_y) \cap D\hat{e}c_C(Q_y) \neq \varnothing$ is equivalent to $D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y)$.

By Theorem 2, each n-dimensional decoding cone corresponds to a region of the braid arrangement $\mathcal{B}_n$. As noted in Section 2.3 to each $\sigma \in S_n$ there corresponds a region $R_\sigma \in \mathcal{R}(\mathcal{B}_n)$. We can therefore identify every n-dimensional decoding cone with a permutation in S_n.

Example 5. *Consider $\mathbb{R}^3_{\geqslant 0}$. The identity element $1 \in S_3$ corresponds to the cone with ordering $(1 \prec 2 \prec 3)$. The transposition $(13) \in S_3$ corresponds to $(3 \prec 2 \prec 1)$.*

Since decoding depends exclusively on the decoding cone, we can extend the definition of $D\hat{e}c_C$ to permutations in the following way.

Definition 9. *Let $\sigma \in S_n$, $R_\sigma \in \mathcal{R}(\mathcal{B}_n)$ its corresponding decoding cone and $P \in Cha_{n\times m}$ such that $P \in R_\sigma$. We define $D\hat{e}c_C(\sigma) = D\hat{e}c_C(P)$ for every $C \subseteq X$.*

The leading question we posed in the beginning of this section can now be restated in terms of permutation groups as follows:

Given two permutations $\sigma, \phi \in S_n$, what is the probability that $D\hat{e}c_C(\sigma) = D\hat{e}c_C(\phi)$ if $C \subseteq X$ is chosen with uniform distribution?

We will solve this by basic counting.

Definition 10. *Let $\sigma, \phi \in S_n$. We denote by $S(\sigma, \phi)$ the number of codes C for which $D\hat{e}c_C(\sigma) = D\hat{e}c_C(\phi)$.*

This function is permutation invariant.

Proposition 8. *Let $\sigma, \phi, \tau \in S_n$. Then, $S(\tau \circ \sigma, \tau \circ \phi) = S(\sigma, \phi)$.*

Proof. This follows from the fact that if you permute the rows of a channel, the same permutation on a maximum likelihood decoder of it will yield a maximum likelihood decoder of the permuted channel. □

Thus, we can define $S(\sigma) = S(1, \sigma)$ and then $S(\sigma, \phi) = S(\phi^{-1} \circ \sigma)$.

We now show how to compute this function.

Theorem 4. *Let $\sigma \in S_n$ and let us define $f_i(\sigma) = \sum_{j=i+1}^{n} [\sigma^{-1}(i) \leqslant \sigma^{-1}(j)]$. Then,*

$$S(\sigma) = \sum_{i=1}^{n} 2^{f_i(\sigma)}.$$

Proof. We want to count how many codes such that $D\hat{e}c_C(1) = D\hat{e}c_C(\sigma)$. The identity element represents $(1 \prec 2 \prec \ldots \prec n)$ and σ represents $(\sigma^{-1}(1) \prec \sigma^{-1}(2) \prec \ldots \prec \sigma^{-1}(n))$.

First consider codes C such that $x_1 \in C$. The identity element 1 will decode any one of these as x_1. Thus $D\hat{e}c_C(1) = D\hat{e}c_C(\sigma)$ if and only if σ also decodes as x_1. For this to happen, C can only contain elements x_i such that $\sigma^{-1}(i) \leqslant \sigma^{-1}(j)$. But $f_1(\sigma)$ counts precisely how many of these exist. So the total number of codes satisfying $x_1 \in C$ and $D\hat{e}c_C(1) = D\hat{e}c_C(\sigma)$ is $2^{f_1(\sigma)}$.

Now consider codes C such that $x_1 \notin C$ and $x_2 \in C$. The same reasoning yields the total number of codes satisfying $x_1 \notin C$, $x_2 \in C$ and $D\hat{e}c_C(1) = D\hat{e}c_C(\sigma)$ as $2^{f_2(\sigma)}$.

Continuing with the same argument yields our result. □

The next theorem answers the question posed in the beginning of this section.

Theorem 5. *Let $\sigma, \phi \in S_n$. If a code $C \subseteq X$ is picked uniformly distributed from the space of all codes, then $Pr(D\hat{e}c_C(\sigma) = D\hat{e}c_C(\phi)) = \dfrac{S(\phi^{-1} \circ \sigma)}{2^n - 1}$.*

Proof. By definition, $S(\phi^{-1} \circ \sigma)$ counts the number of codes such that $D\hat{e}c_C(\sigma) = D\hat{e}c_C(\phi)$. Elementary probability says we must divide this by the total number of codes. □

With this we can define a distance between permutations.

Definition 11. *The decoding distance between two permutations $\sigma, \phi \in S_n$ is*

$$d_{dec}(\sigma, \phi) = 1 - Pr(\ \hat{Dec}_C(\sigma) = \hat{Dec}_C(\phi)\ )$$

In the context of the braid arrangement there exists already a natural distance between permutations. It is known as the Kendall tau distance [24], which we denote by $d_\tau(\sigma, \phi)$, and is defined as the minimum number of adjacent permutations $\tau_1, \tau_2, \ldots, \tau_{d_\tau(\sigma,\phi)}$ so that $\phi = \sigma \circ \tau_1 \circ \tau_2 \circ \ldots \circ \tau_{d_\tau(\sigma,\phi)}$.

Consider the graph whose vertices are the regions of the braid arrangement and such that two vertices share an edge if their corresponding regions are adjacent to each other (so that each edge corresponds to a hyperplane). Then, the Kendall Tau Distance is the shortest path distance of the graph.

In technical terms: if $\sigma, \tau \in S_n$ where $\tau = (r, r+1)$ and 1 is the identity in S_n, then

$$d_\tau(1, \tau \circ \sigma) - d_\tau(1, \sigma) \begin{cases} -1 & if\ \ \sigma^{-1}(r) < \sigma^{-1}(r+1) \\ 1 & if\ \ \sigma^{-1}(r) > \sigma^{-1}(r+1) \end{cases}$$

We now show that the decoding distance behaves as a weighed version of the Kendall tau distance.

Theorem 6. *Let $\sigma, \tau \in S_n$ where $\tau = (r, r+1)$. Then,*

$$S(\tau \circ \sigma) - S(\sigma) = \begin{cases} -2^{f_r(\sigma)-1} & if\ \ \sigma^{-1}(r) < \sigma^{-1}(r+1) \\ 2^{f_{r+1}(\sigma)} & if\ \ \sigma^{-1}(r) > \sigma^{-1}(r+1) \end{cases}$$

Proof. Since $(\tau\circ\sigma)^{-1}(r) = \sigma^{-1}(r+1)$ and $(\tau\circ\sigma)^{-1}(j) = \begin{cases} \sigma^{-1}(r) & if\ \ j = r+1 \\ \sigma^{-1}(j) & if\ \ j > r+1 \end{cases}$ it follows that

$$\begin{aligned} f_r(\tau \circ \sigma) &= \sum_{j=r+1}^{n} [(\tau \circ \sigma)^{-1}(r) \leqslant (\tau \circ \sigma)^{-1}(j)] \\ &= [\sigma^{-1}(r+1) \leqslant \sigma^{-1}(r)] + f_{r+1}(\sigma) \end{aligned}$$

Since $(\tau\circ\sigma)^{-1}(r+1) = \sigma^{-1}(r)$ and $r+1 < j \Rightarrow (\tau\circ\sigma)^{-1}(j) = \sigma^{-1}(j)$ it follows that

$$\begin{aligned} f_{r+1}(\tau \circ \sigma) &= \sum_{j=r+2}^{n} [(\tau \circ \sigma)^{-1}(r+1) \leqslant (\tau \circ \sigma)^{-1}(j)] + [\sigma^{-1}(r) \leqslant \sigma^{-1}(r+1)] - [\sigma^{-1}(r) \leqslant \sigma^{-1}(r+1)] \\ &= f_r(\sigma) - [\sigma^{-1}(r) \leqslant \sigma^{-1}(r+1)] \end{aligned}$$

Thus, we have

$$\begin{aligned} S(\tau \circ \sigma) &= \sum_{i=1}^{r-1} 2^{f_i(\sigma)} + 2^{f_r(\tau\circ\sigma)} + 2^{f_{r+1}(\tau\circ\sigma)} + \sum_{i=r+2}^{n} 2^{f_i(\sigma)} \\ &= S(\sigma) + 2^{f_r(\sigma)}(2^{-[\sigma^{-1}(r)\leqslant\sigma^{-1}(r+1)]} - 1) + 2^{f_{r+1}(\sigma)}(2^{[\sigma^{-1}(r+1)\leqslant\sigma^{-1}(r)]} - 1). \end{aligned}$$

□

3.5 A Distance Between Channels

In this section we extend the results of the previous one to define a distance (in some sense) between Channels. As in the last section we will only consider the case where the decoding cones are full dimensional.

We could define a distance by setting $d(P,Q) = 1 - Pr(\ D\hat{e}c_C(P) = D\hat{e}c_C(\phi)\ )$. But we would not be using any information on the output message.

To illustrate this consider three channels $P, Q, R \in Cha_3$ such that

$$O^-P = \begin{pmatrix} 1 & 3 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix} \quad O^-Q = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix} \quad O^-R = \begin{pmatrix} 2 & 3 & 2 \\ 1 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

One can check by doing all possible computations that $d(P,Q) = d(P,R) = d(Q,R) = \frac{4}{7}$. But Q differs from P in only one position of a single column, while R differs from P in one position in two different columns. If y_1 or y_2 (the output messages corresponding, respectively, to the first and second columns) is received P and Q are essentially the same channel. Intuitively, we expect Q to be closer to P than R is.

If we assume that the transmission is made through the channel P, and denote by Q_y the column corresponding to the received message y in Q, we can calculate $Pr(\ D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y)\ )$, the probability that both decoders will be equal when a message y is received.

Theorem 7. *Let $P, Q \in Ch_{n\times m}$ and $\sigma, \phi \in S_n^m$ be such that σ_i (ϕ_i) corresponds to the ordering in the i-th column of O^-P (O^-Q). Suppose that the channel being used is P. If a code $C \subseteq X$ is picked uniformly distributed from the space of all codes, then*

$$Pr(\ D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y)\ ) = \frac{1}{n(2^n-1)} \sum_{i=1}^{m} S(\sigma_i, \phi_i)\, \|P_i\|_1$$

where $\|P_i\|_1 := \sum_{j=1}^{n} P_{ji}$ is the 1-norm of the i-th column of P.

Proof.

$$\begin{aligned} Pr(\ D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y)\ ) &= \sum_{i=1}^{m} Pr(\ D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y)|\ y_i \text{ received})Pr(y_i \text{ received}) \\ &= \sum_{i=1}^{m} \frac{S(\sigma_i, \phi_i)}{2^n - 1} \sum_{j=1}^{n} Pr(y_i \text{ received } | x_j \text{ sent})Pr(x_j \text{ sent}) \\ &= \sum_{i=1}^{m} \frac{S(\sigma_i, \phi_i)}{2^n - 1} \|P_i\|_1 \frac{1}{n} \end{aligned}$$

□

In the hypothesis of Theorem 7 we assume that one of the channels is the correct one. This occurs because the expression depends on the probability of receiving y which may not coincide for different channels.

Example 6. *Suppose a channel* $P = \begin{pmatrix} \frac{5}{8} & \frac{1}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{5}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{2}{8} & \frac{5}{8} \end{pmatrix}$ *is used for transmission and* $Q, R \in Cha_3$ *are such that*

$$O^-Q = \begin{pmatrix} 1 & 3 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix} \text{ and } O^-R = \begin{pmatrix} 2 & 3 & 2 \\ 1 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

Then, by Theorem 7,

$$Pr(D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y) = \frac{1}{21}(7 + 7 + 4) = \frac{6}{7}$$

and

$$Pr(D\hat{e}c_C(P_y) = D\hat{e}c_C(R_y)) = \frac{1}{21}(5 + 7 + 4) = \frac{16}{21}.$$

We note that this difference is, intuitively, compatible with the simple observation that Q differs from P in only one position of a single column, while R differs from P in one position in two different columns.

We recall that in Theorem 7 we considered two channels, P and Q, with P having a distinguished role: we assumed that the transmition is made over P. Out of it, we can get a way to measure the distances between an arbritaty channel Q and the actual channel P, as follows.

Definition 12. *Let $P, Q \in Ch_{n\times m}$ and assume that P is the channel being used. The decoding distance from Q to P is given by*

$$d^P_{dec}(Q) = 1 - Pr(\ D\hat{e}c_C(P_y) = D\hat{e}c_C(Q_y)\ ).$$

Let P, Q and R be as in Example 6. By Definition 12, $d^P_{dec}(Q) = \frac{1}{7} < \frac{5}{21} = d^P_{dec}(R)$. Thus, as we would expect, Q is closer to P than R is.

4 Channel Metrization

We come back now to the line of thought we left at the end of Section 3.3.

In this chapter we consider the problem of determining if a channel is matched to a metric, or in other words, if a channel is metrizable (in analogy to metrizable spaces in topology). This problem was first posed by Massey in [28]. Since then, this problem has been very little explored. In 1980, a first approach concerning classical additive metrics was explored by Seguin (see [36][1]). The problem rested untouched until 2016, when a sequence of works established the metrization of the Z-channels and the binary asymmetric channels (see [15][2], [33][3] and [34][4]).

When metrization occurs, it gives more structure to the channel, and it might be useful for, among other things, the construction of efficient codes and decoding algorithms. This is the reason why metric invariants - such as minimal distance, packing radius, perfect codes, MDS codes, etc - are considered to be important and taken for granted in coding theory.

In Section 4.1 we give both a geometrical and graph theoretical characterization of channel metrization.

In Section 4.2 present an algorithm which determines if a channel is metrizable.

4.1 A Characterization of Channel Metrization

If $d \in Dis_n$ is a distance, it was proved in [15] that it is always matchable to some channel. One may think on the reciprocal question: Given a channel $P \in Cha_n$, under which condition is P metrizable? Or in other words, how do we determine if there exists a metric d matched to P?

We first note that in Example 2 it is shown that every semimetric is decoding-equivalent to a metric. Thus, matching a channel to a metric is equivalento to matching it to a semimetric.

The only difference between maximum likelihhod decoding and minimum distance decoding is that in the former case we look for the largest entry in the column while in the latter we look for the smallest one. Thus, the problem of channel metrization is

[1] Séguin considers families of metrics that are defined over an alphabet and which extend additively over the coordinates, which include the Hamming and Lee metrics.

[2] Walker and Firer proved that a metric can be matched to the Z-Channel, which in a sense is the most anti-symmetrical of all channels.

[3] Poplawski shows a weak form of metrization of the binary asymmetric channels.

[4] Qureshi proves the metrization conjecture for the binary asymmetric channels.

equivalent to determining if for a given channel $P \in Cha_n$ there exists a distance $d \in Sem_n$ such that $O^- P = O^+ d$. If we apply a convenient transformation to d we can show the following.

Proposition 9. *A channel $P \in Cha_n$ is metrizable if and only if there exists $d \in Sem_n$ such that $O^- P = O^- d$.*

Proof. We know that P is metrizable if and only if there exists $d' \in Sem_n$ such that $O^- P = O^+ d'$.

Let d be defined such that $d(x, y) = \dfrac{[x \neq y]}{d'(x, y)}$. The result follows from the fact that $O^+ d' = O^- d$. □

From this we can give a geometric characterization of channel metrization.

Theorem 8. *A channel $P \in Cha_n$ is metrizable if and only if $Cone(P) \cap Sem_n \neq \varnothing$.*

Proof. By proposition 9, P is metrizable if and only if there exists $d \in Sem_n$ such that $O^- P = O^- d$. But this occurs if and only if P and d belong to the same decoding cone. Thus $d \in Cone(P)$. □

We will give another characterization in term of the mixed graph associated to $Order(P)$. This will depend on the following definition.

Definition 13. *A contradiction cycle in a mixed graph G is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that:*

1. *$v_k = v_1$,*
2. *either $v_i \simeq v_j$ or $v_i \prec v_j$,*
3. *for some i, $v_i \prec v_j$.*

The idea here is that if a mixed graph has a contradiction cycle it will not be a preorder since it will not satisfy condition 4 of the definition.

We now characterize channel metrization in terms of mixed graphs.

Theorem 9. *Let $P \in Cha_n$. Then, P is metrizable if and only if $Order(P)$ with additional undirected edges joining (i, j) to (j, i) for every $i, j \in [n]$ satisfies:*

1. *it has no contradiction cycles,*
2. *for every $i \in [n]$, there are no undirected edges connected to (i, i),*
3. *for every $i \in [n]$, there are no directed edges pointing to (i, i).*

Proof. We denote by G the mixed graph $Order(P)$ with additional undirected edges joining (i, j) to (j, i) for every $i, j \in [n]$.

If P is metrizable then, by Theorem 8, there exists $d \in Sem_n$ such that $Order(d) = Order(P)$.

Property 1 follows from the symmetry property of distances making G a preorder which, therefore, contains no contradiction cycles.

Properties 2 and 3 follow from the identity of indiscernibles.

Now, let $P \in Cha_n$ and suppose that G satisfies the three properties enumerated above. We show how to construct a distance $d \in Sem_n$ such that $Order(P) = Order(d)$.

By Definition 8, each column of P has a weak ordering associated to it. Thus, we have n chains of $n - 1$ inequalities. Properties 2 and 3 guarantee that only $d(i, i) = 0$ as should be for any distance.

To construct our distance we do the following steps:

1. We take the first chain and set arbitrary values for the distances with the condition that the inequalities hold true.
2. We then set the same values to their corresponding symmetric term, i.e. $d(i, j) = d(j, i)$;.
3. We continue to do this for the next chain until we have assigned a value to every distance and have therefore found a distance matched to our channel.

The only way for this procedure not to work is if some distance cannot have a value assigned to them. But if this happens we will have found a contradiction cycle, contradicting property 1. □

Example 7. *Consider the channel P and its order matrix*

$$P = \begin{pmatrix} \frac{5}{8} & \frac{1}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{5}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{2}{8} & \frac{5}{8} \end{pmatrix} \quad \textit{and} \quad O^{-}P = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

Figure 7 shows that G_P contains a cycle and therefore P is not metrizable.

4.2 An Algorithm for Channel Metrization

In this section we give an algorithm for determining if a channel is metrizable. Our algorithm is essentially the second half of the proof of Theorem 9.

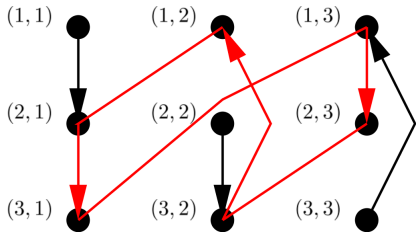

Figure 5 – The mixed graph G_P.

Algorithm 1. ***Input:*** *A channel $P \in Cha_n$.*

Output: *A distance $d \in Sem_n$ matched to P or a chain of inequalieties showing that no such distance exists.*

We have n chains of inequalities, each corresponding to a certain column. The smallest element of each chain must be $d(i,i)$ which we set to equal zero. Then:

1. *We take the first chain and set arbitrary values for the distances with the condition that the inequalities hold true;*
2. *We then set the same values to their corresponding symmetric term, i.e. $d(i,j) = d(j,i)$;*
3. *We continue to do this until we have assigned a value to every distance and have therefore found a distance matched to our channel;*
   or,

3'. *Find that some distance cannot have a valued assigned to it, and thus there is no distance matched for this channel. In this case a contradiction cycle has been found and we use it as an output.*

We illustrate the use of the algorithms in the following two examples.

Example 8. *Let* $M = \begin{pmatrix} \frac{5}{8} & \frac{1}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{5}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{2}{8} & \frac{5}{8} \end{pmatrix}$. *Then,* $O^- M = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{pmatrix}$.

We have the following three chains of inequalities (corresponding to the columns):

$$0 = d(1,1) < d(1,2) < d(1,3)$$

$$0 = d(2,2) < d(2,3) < d(2,1)$$

$$0 = d(3,3) < d(3,1) < d(3,2)$$

We set arbitrary values to the elements in the first column and the same to their symmetric counterparts (since a distance is symmetric).

$$0 = d(1,1) < 1 = d(1,2) < 2 = d(1,3)$$

$$0 = d(2,2) < d(2,3) < 1 = d(2,1)$$

$$0 = d(3,3) < 2 = d(3,1) < d(3,2)$$

In the next step we must set an arbitrary value to $d(2,3)$ *but it is impossible to do this since it must satisfy*

$$2 = d(3,1) < d(2,3) < 1 = d(2,1).$$

Therefore, M *is not metrizable since the following is not possible*

$$d(2,1) < d(3,1) < d(2,3) < d(2,1).$$

Example 9. *Let* $M = \begin{pmatrix} \frac{5}{8} & \frac{3}{16} & \frac{3}{16} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{2}{8} & \frac{5}{8} \end{pmatrix}$. *Then,* $O^{-}M = \begin{pmatrix} 1 & 3 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$.

We have the following three chains of inequalities:

$$0 = d(1,1) < d(1,2) < d(1,3)$$
$$0 = d(2,2) < d(2,3) < d(2,1)$$
$$0 = d(3,3) < d(3,2) < d(3,1)$$

We set arbitrary values to the first chain and to their symmetric counterparts.

$$0 = d(1,1) < 1 = d(1,2) < 2 = d(1,3)$$

$$0 = d(2,2) < d(2,3) < 1 = d(2,1)$$

$$0 = d(3,3) < d(3,2) < d(3,1) = 2$$

We do the same for the second chain.

$$0 = d(1,1) < 1 = d(1,2) < 2 = d(1,3)$$
$$0 = d(2,2) < d(2,3) = \frac{1}{2} < 1 = d(2,1)$$
$$0 = d(3,3) < d(3,2) = \frac{1}{2} < d(3,1) = 2$$

We were able to set values to all the distances. Therefore, M is matched to the following distance:

$$d = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & \frac{1}{2} \\ 2 & \frac{1}{2} & 0 \end{pmatrix}.$$

Note that this distance is not a metric since

$$2 = d(1,3) > d(1,2) + d(2,3) = 1 + \frac{1}{2}.$$

Since $O^{-}M$ has ones only in the diagonal we can apply the transformation of Example 2 to get a metric matched to the channel. In this case the metric

$$d_2 = \begin{pmatrix} 0 & \frac{3}{2} & 2 \\ \frac{3}{2} & 0 & \frac{5}{4} \\ 2 & \frac{5}{4} & 0 \end{pmatrix}$$

is matched to the channel M.

In terms of algorithmic complexity, if our matrix is $n \times n$ we have at most n^2 elements to set and for each of these we make at most $2n$ comparisons (each chain of inequalities has size n), giving a trivial upper bound of $O(n^3)$ on the number of operations to be performed.

Unfortunately, there is an undesirable situation one can not avoid: in applications, considering for example block codes of length r, the size $n(r)$ of the matrix is usually exponential on r.

5 Hamming Cube Embeddings

It is a common theme throughout mathematics to study under which conditions some complicated structure can be embedded into a simpler one. In the area of finite distances one of the simpler structures is the Hamming distance on the hypercube (which we refer to as the Hamming cube).

Isometric Hamming cube embedding is an active area of research with many applications [7, 9] both within mathematics (e.g. geometry of numbers, analysis, probability theory) and to other sciences (e.g. computer science, statistical physics, biology). Determining if an embedding exists is an NP-Hard problem [4]. In coding theory, specific instances were studied in [3, 17].

In this chapter we show that, in the context of coding theory, every semimetric is isometrically embeddable, up to decoding equivalence, into the Hamming Cube. Thus, for decoding purposes, the Hamming distance is universal. We also study the minimum dimension of such an embedding.

In Section 5.1 we generalize the notion of intersection patterns to what we call set patterns. Solving these types of problems is equivalent to finding Hamming cube embeddings.

In Section 5.2 we show the universality of the Hamming metric, i.e., every semimetric is isometrically embeddable, up to decoding equivalence, into the Hamming Cube.

In Section 5.3 we show that finding the minimum dimension of an embedding is a linear programming problem. We present an algorithm which gives us an approximate solution and give bounds on the optimal solution.

5.1 Set Patterns

In [6], it is shown that isometrically embedding a distance into the Hamming cube is equivalent to solving a problem of the following type: given an $n \times n$ matrix $A = (a_{ij})$, decide whether there exists sets $S_1, S_2, \ldots, S_n$ such that $|S_i \cap S_j| = a_{ij}$ for every $i, j \in [n]$.

The matrix A is known as an *intersection pattern*, and when such sets exist, the intersection pattern is said to be *realizable*.

Determining if an intersection problem is realizable is NP-complete [4] and, therefore, so is determining if a distance is isometrically embeddable into the Hamming

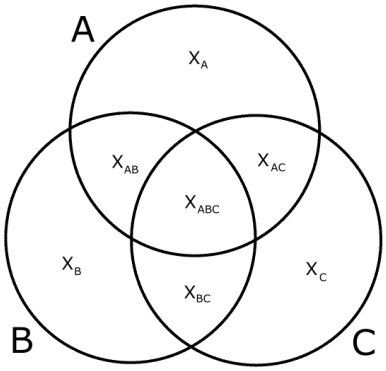

Figure 6 – Minterms of the family $\mathcal{F} = \{A, B, C\}$.

cube.

To prove our results we will need to generalize on the notion of intersection patterns by defining what we call *set patterns*.

We tacitly assume that all subsets I considered in the sequel are nonempty.

Definition 14. *Given a finite family of sets $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$ a* minterm[1] *is a set $X_I = \{a \in \cup_{i=1}^{n} A_i : i \in I \Rightarrow a \in A_i, i \notin I \Rightarrow a \notin A_i, \forall i \in [n]\}$, for every subset $I \subseteq [n]$. The cardinalities of the minterms are denoted by lowercase letters $x_I = |X_I|$.*

The minterms are the disjoint components of the Venn diagram of the sets.

We now present the main notion of this section.

Definition 15. *A* set pattern *is a triple (G, c, n) where n is a positive integer, $G : \mathbb{R}^{2^n-1} \to \mathbb{R}^m$, and $c \in \mathbb{R}^m$. An $x \in \mathbb{R}^{2^n-1}$ is called a* solution *of the*[2] *set pattern if $G(x) = c$.*

If there exists a finite family of sets $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$ such that the cardinalities of the minterms are a solution of the set pattern, we say that $\mathcal{F}$ is a realization *of the set pattern and that the set pattern is* realizable.

A set pattern is, essentially, a system of equations for which integer solutions correspond to a family of sets satisfying the pattern imposed by the equations.

Proposition 10. *A set pattern (G, c, n) is realizable if and only if there exists $x \in \mathbb{Z}_{\geqslant 0}^{2^n-1}$ such that $G(x) = c$.*

Proof. It follows directly from the definition. □

Example 10. *Does there exist sets A_1, A_2, A_3 such that $|A_1 \triangle A_2| = 3$, $|A_3|^{|A_1|} = 27$ and $|A_1 \cap A_2|^2 = 9$?*

[1] This term is taken from Boolean algebra.

[2] We will use subsets as indexes and leave unspecified, but assume as given, the bijection from $[2^n - 1]$ to $2^{[n]} - \varnothing$.

This problem is equivalent to the realizability of the following set pattern:

$$\begin{cases} G_1(x) = x_1 + x_2 + x_{13} + x_{23} & = 3 & = c_1 \\ G_2(x) = (x_3 + x_{13} + x_{23} + x_{123})^{x_1+x_{12}+x_{13}+x_{123}} & = 27 & = c_2 \\ G_3(x) = (x_{12} + x_{123})^2 & = 9 & = c_3 \end{cases}$$

Since $(x_1, x_2, x_3, x_{12}, x_{13}, x_{23}, x_{123}) = (2, 0, 1, 2, 0, 1, 1) \in \mathbb{Z}^7_{\geqslant 0}$ *is a solution, it follows from Proposition 10 that such sets exist.*

We are only interested in set patterns which correspond to intersections and symmetric differences since these will be used for our Hamming embeddings. We must generalize these functions to $\mathbb{R}^{2^n-1}$.

We start by generalizing set intersections.

Definition 16. *Let* $J \subseteq [n]$. *The* J-wise intersection function *is defined as* $\Cap_J : \mathbb{R}^{2^n-1} \to \mathbb{R}$ *such that*

$$\Cap_J(x) = \sum_{J \subseteq I \subseteq [n]} x_I.$$

The J*-wise intersection functions, where* $|J| \leqslant k$, *is denoted by* $\boldsymbol{\Cap_k} = (\Cap_{J\subseteq[n]})_{|J|\leqslant k}$.

It is easy to see that J-wise intersections are linear and that, moreover, $\boldsymbol{\Cap_n}$ is a linear automorphism[3], and thus, has a unique solution.

Example 11. *Does there exist sets* A_1, A_2, A_3 *such that*

$$\begin{array}{lll} |A_1| = 6 & |A_1 \cap A_2| = 6 & |A_1 \cap A_2 \cap A_3| = 4 \\ |A_2| = 9 & |A_1 \cap A_3| = 5 & ? \\ |A_3| = 8 & |A_2 \cap A_3| = 7 & \end{array}$$

This problem is equivalent to the realizability of the following set pattern:

$$\begin{array}{lll} \Cap_1(x) = 6 & \Cap_{12}(x) = 6 & \Cap_{123}(x) = 4 \\ \Cap_2(x) = 9 & \Cap_{13}(x) = 5 & \\ \Cap_3(x) = 8 & \Cap_{23}(x) = 7 & \end{array}$$

or equivalently, $\boldsymbol{\Cap_n}(x) = (6, 9, 8, 6, 5, 7, 4)$. *The unique solution is* $x = (-1, 0, 0, 2, 1, 3, 4) \notin \mathbb{Z}^7_{\geqslant 0}$ *and thus, by Proposition 10, no such sets exist.*

Intersection patterns are a particular case of set patterns of the form $(\boldsymbol{\Cap_2}, c, n)$.

[3] A linear transformation from $\mathbb{R}^{2^n-1}$ to itself.

Given a family $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$, we have the following known relation

$$|\triangle_{i\in I} A_i| = \sum_{l=1}^{|I|} (-2)^{l-1} \sum_{\substack{K\subseteq I \\ |K|=l}} |\cap_J A_{i\in J}|. \tag{5.1}$$

We use equation 5.1 to generalize symmetric differences.

Definition 17. *Let $J \subseteq [n]$. The* J-wise symmetric difference function *is defined as the function $\blacktriangle_J : \mathbb{R}^{2^n-1} \to \mathbb{R}$ such that*

$$\blacktriangle_J(x) = \sum_{l=1}^{|J|} (-2)^{l-1} \sum_{\substack{K\subseteq I \\ |K|=l}} \Cap_K(x)$$

The J-wise symmetric difference functions, where $|J| \leqslant k$, is denoted by $\blacktriangle_{\boldsymbol{k}} = (\blacktriangle_{J\subseteq[n]})_{|J|\leqslant k}$.

It follows directly from the linearity of J-wise intersections that J-wise symmetric differences are linear and that $\blacktriangle_{\boldsymbol{n}}$ is a linear automorphism.

Example 12. *Does there exist sets A_1, A_2, A_3 such that*

$$\begin{array}{lll} |A_1| = 3 & |A_1 \triangle A_2| = 3 & |A_1 \triangle A_2 \triangle A_3| = 3 \\ |A_2| = 2 & |A_1 \triangle A_3| = 3 & ? \\ |A_3| = 1 & |A_2 \triangle A_3| = 2 & \end{array}$$

This problem is equivalent to the realizability of $\blacktriangle_{\boldsymbol{n}}(x) = (3, 2, 1, 3, 3, 2, 3)$.

By Definition 17, we can recursively calculate the J-wise intersections and show that the problem is equivalent to the realizability of $\Cap_{\boldsymbol{n}}(x) = (3, 2, 1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{4})$, which has as a unique solution $x = \frac{1}{4}(7, 3, 1, 3, 1, 1, 1) \notin \mathbb{Z}^7_{\geqslant 0}$.

Thus, no such sets exist.

We denote by $\overrightarrow{\mathbf{1}}$ the vector $(1, 1, \ldots, 1)$ with all entries equal to one.

The following Lemma will be essential for proving Theorem 10.

Lemma 1. *It holds that $\blacktriangle_{\boldsymbol{n}}(\overrightarrow{\mathbf{1}}) = 2^{n-1}\overrightarrow{\mathbf{1}}$.*

Proof. We start by calculating $\Cap_J(\overrightarrow{\mathbf{1}}) = \sum_{I \in [n]} [J \subseteq I] = 2^{n-|J|}$. Thus,

$$\begin{aligned}
\blacktriangle_J(\overrightarrow{\mathbf{1}}) &= \sum_{l=1}^{|I|}(-2)^{l-1} \sum_{\substack{K \subseteq I \\ |K|=l}} \Cap_K(\overrightarrow{\mathbf{1}}) = \sum_{l=1}^{|I|}(-2)^{l-1} \sum_{\substack{K \subseteq I \\ |K|=l}} 2^{n-|K|} \\
&= \sum_{l=1}^{|I|}(-2)^{l-1}\binom{|I|}{l}2^{n-l} = 2^{n-1}\sum_{l=1}^{|I|}\binom{|I|}{l}(-1)^{l-1} \\
&= 2^{n-1},
\end{aligned}$$

where in the last equality we use the identity $\sum_{i=1}^{k}\binom{k}{i}(-1)^{i-1} = 1$.

□

We are now ready to prove the main result of this section.

Theorem 10. *Given $c \in \mathbb{Q}_+^{2^n-1}$, there exists positive integers m and k such that the set pattern $(\blacktriangle_{\boldsymbol{n}}, (mc + \overrightarrow{\boldsymbol{k}}), n)$ is realizable.*

Proof. Let x be the solution of $(\blacktriangle_{\boldsymbol{n}}, c, n)$. By definitions 16 and 17, the rationality of the x_I follows from that of the c_I.

Let m be the least common multiple of the divisors of all the x_I, $r = |\min_{I \subseteq [n]}\{mx_I\}|$, and $k = r2^{n-1}$. Then, for every $I \subseteq [n]$, by definition of m, we have that $mx_I + r$ is an integer and that it is non-negative, since

$$\min_{I \subseteq [n]}\{mx_I + r\} = \min_{I \subseteq [n]}\{mx_I\} + r \geqslant 0.$$

By linearity and Lemma 1, it follows that

$$\begin{aligned}
\blacktriangle_{\boldsymbol{n}}(mx + \overrightarrow{\boldsymbol{r}}) &= m\blacktriangle_{\boldsymbol{n}}(x) + \blacktriangle_{\boldsymbol{n}}(\overrightarrow{\boldsymbol{r}}) \\
&= mc + 2^{n-1}\overrightarrow{\boldsymbol{r}} \\
&= mc + \overrightarrow{\boldsymbol{k}}.
\end{aligned}$$

Thus, by Proposition 10, $(\blacktriangle_{\boldsymbol{n}}, (mc + \overrightarrow{\boldsymbol{k}}), n)$ is realizable. □

Note that $mc + \overrightarrow{\boldsymbol{k}}$ has the same weak ordering as c. This is important since, as seen in Theorem 1, two distances are decoding equivalent if they have the same weak ordering. It is in this way that we will use Theorem 10 to prove Theorems 11 and 12.

Example 13. *Let us apply Theorem 10 to Example 12.*

We saw that $\blacktriangle_{\boldsymbol{n}}(x) = (3,2,1,3,3,2,3)$ *has unique solution* $x = \frac{1}{4}(7,3,1,3,1,1,1)$. *From the proof of Theorem 10 it follows that if we take* $m = 2$ *and* $k = \frac{1}{4}$, *then* $\blacktriangle_{\boldsymbol{n}}(x) = 2(3,2,1,3,3,2,3) + \frac{1}{4}\overrightarrow{\mathbf{1}}$ *is realizable. Indeed one can calculate that its solution is* $x = (4,2,1,2,1,1,1)$.

Both $\frac{1}{4}(7,3,1,3,1,1,1)$ *and* $(4,2,1,2,1,1,1)$ *have the same weak ordering.*

5.2 Embedding Distances into the Hamming Cube

Embedding distances isometrically into the Hamming cube is an area of its own [7]. Determining if it is possible for a given distance is NP-Hard [4].

We prove that any semimetric is decoding equivalent to a distance which is isometrically embeddable into the Hamming cube. If, in addition, the semimetric is translation invariant over $\mathbb{F}_2^n$, the embedding is a linear function.

We first note that there is a weight preserving bijection between the n-dimensional Hamming cube H^n, and the subsets of $[n]$, $2^{[n]}$ given by

$$supp : H^n \to 2^{[n]}$$

where $supp(x) = \{i : x_i \neq 0\}$.

This function satisfies the following properties:

1. $supp(x + y) = supp(x) \triangle supp(y)$
2. $\omega_H(x) = |supp(x)|$.

Thus, isometrically embedding a distance d over $[n]$ into the Hamming cube is equivalent to determining if $(\blacktriangle_{\mathbf{2}}, \delta, n-1)$ is realizable, where

$$\delta_{ij} = d(i,j), \quad i,j \in [n-1]$$

$$\delta_i = d(i,n), \quad i \in [n-1].$$

By Definition 17 this corresponds to the intersection pattern $(\Cap_{\mathbf{2}}, c, n-1)$:

$$c_{ij} = \frac{1}{2}(d(i,n) + d(j,n) - d(i,j)), \quad i,j \in [n-1]$$

$$c_i = d(i,n), \quad i,j \in [n-1].$$

This relation between intersection patterns and Hamming cube embeddings was first pointed out by Deza in [6].

We are interested in embedding up to decoding equivalence. We will first consider the case of translation invariant semimetrics over $\mathbb{F}_2^n$. This will follow directly from Theorem 10. Since any semimetric can be seen as translation invariant by adding dummy variables, the general case will follow as a consequence.

As said earlier, we always assume $I \subseteq [n]$ to be nonempty.

Theorem 11. *Let d_1 be a translation invariant semimetric over $\mathbb{F}_2^n$ with weight ω_1. Then there exists a translation invariant semimetric d_2, with weight ω_2 which is decoding equivalent to d_1 and is linearly embeddable into the Hamming cube.*

Proof. Denote by $\{e_1, e_2, \cdots, e_n\}$ the standard basis of $\mathbb{F}_2^n$.

Let, for every $I \subseteq [n]$,

$$\delta_I = \omega_1 \left(\sum_{i \in I} e_i \right).$$

Without loss of generality we can assume that $\delta_I \in \mathbb{Q}_+$, since, by Theorem 1, only the order relation between the values matters.

Again by Theorem 1, for any given $m, k \in \mathbb{Z}_+$ if , for every $I \subseteq [n]$,

$$w_2 \left(\sum_{i \in I} e_i \right) = m\delta_I + k,$$

then $d_1 \sim d_2$ since the ordering of the distances is preserved.

Theorem 10 ensures that there exists $m, k \in \mathbb{Z}_+$ such that $(\blacktriangle_{\boldsymbol{n}}, m\delta + \vec{\boldsymbol{k}}, n)$ is realizable. Thus, there exists a family of sets, $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$, such that, for every $I \subseteq [n]$, $|\triangle_{i \in I} A_i| = m\delta_I + k$.

Let $N = |\cup_i A_i|$ and $f : \mathbb{F}_2^n \to 2^N$ be such that $f(e_i) = A_i$. Then, $supp^{-1} \circ f$ is a linear embedding from $(\mathbb{F}_2^n, d_2)$ (where d_2 is decoding equivalent to d_1) into the N dimensional Hamming cube. The requirement that d_1 must be a semimetric is needed for f to be injective. □

We now prove the result for any semimetric.

Theorem 12. *Let d_1 be a semimetric over $[n]$. Then there exists a distance d_2 such that $d_1 \sim d_2$ and d_2 is isometrically embeddable into the Hamming cube.*

Proof. Let, for every $I \subseteq [n]$,

$$\delta_I = \begin{cases} d(i,j) & \text{if} \quad I = \{i,j\} \\ 1 & \text{otherwise} \end{cases}$$

By adding these dummy variables, we can now apply Theorem 11. □

We now show an example of an embedding of a translation invariant metric into the Hamming cube, using the methods described in Theorem 11.

Example 14. *Consider the following translation invariant metric d over $\mathbb{F}_2^3$, with weight ω.*

$$\begin{array}{lll} \omega(001) = 3 & \omega(011) = 3 & \omega(111) = 3 \\ \omega(010) = 2 & \omega(101) = 3 & \\ \omega(100) = 1 & \omega(110) = 2 & \end{array}$$

This corresponds to the following set pattern (which appears in Example 13).

$$\begin{array}{lll} \blacktriangle_1(x) = 3 & \blacktriangle_{12}(x) = 3 & \blacktriangle_{123}(x) = 3 \\ \blacktriangle_2(x) = 2 & \blacktriangle_{13}(x) = 3 & \\ \blacktriangle_3(x) = 1 & \blacktriangle_{23}(x) = 2 & \end{array}$$

Using Equation 5.1 recursively we get the equivalent set pattern.

$$\begin{array}{lll} \Cap_1(x) = 3 & \Cap_{12}(x) = 1 & \Cap_{123}(x) = \dfrac{1}{4} \\ \Cap_2(x) = 2 & \Cap_{13}(x) = \dfrac{1}{2} & \\ \Cap_3(x) = 1 & \Cap_{23}(x) = \dfrac{1}{2} & \end{array}$$

This solution is given by

$$\begin{array}{lll} x_1 = \dfrac{7}{4} & x_{12} = \dfrac{3}{4} & x_{123} = \dfrac{1}{4} \\ x_2 = \dfrac{3}{4} & x_{13} = \dfrac{1}{4} & \\ x_3 = \dfrac{1}{4} & x_{23} = \dfrac{1}{4} & \end{array}$$

Taking $x' = 2x + \dfrac{1}{4}\overrightarrow{\mathbf{1}}$, by Theorem 10, $(\blacktriangle_{\boldsymbol{n}}, \blacktriangle_{\boldsymbol{n}}(x'), n)$ is realizable with

$$\begin{array}{lll} x'_1 = 4 & x'_{12} = 2 & x'_{123} = 1 \\ x'_2 = 2 & x'_{13} = 1 & \\ x'_3 = 1 & x'_{23} = 1 & \end{array}$$

By Theorem 11, this corresponds to the linear embedding, $f : \mathbb{F}_2^3 \mapsto H^{12}$

$$f(100) = 111111110000$$

$$f(010) = 000001111110$$

$$f(001) = 000011000011$$

and decoding in $(\mathbb{F}_2^3, d)$ *is equivalent to decoding in the image of* f *in* (H^{12}, d_H).

5.3 Optimizing Hamming Cube Embeddings

In this section we present an algorithm which given a translation invariant semimetric d finds an upper bound on the minimum dimension of a Hamming cube embedding, denoted by $dim_H(d)$ and give general upper and lower bounds for this value.

A translation invariant semimetric d over $\mathbb{F}_2^n$ is a symmetric $2^n \times 2^n$-matrix with zeros only on the diagonal which is completely determined by the elements in its first column (by translation invariance).

The space of all such semimetrics is isomorphic to $\mathbb{R}_{>0}^{2^n-1}$ and, from now on, we identify these two spaces. As an index to this space we will use the subsets of $[2^n - 1]$ minus the empty set ordered lexicographically.

As an example, we identify the semimetric $\begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 3 & 1 & 2 & 0 \end{pmatrix}$ with $(2, 1, 3)$, i.e. the first column with the zero omitted.

Throughout the text we denote $2^n - 1$ by N and refer to a translation invariant semimetric over $\mathbb{F}_2^n$ just by the term semimetric (as previously stated).

Determining if there exists a Hamming embedding of $d \in \mathbb{R}_{>0}^N$ is equivalent to finding an $x \in \mathbb{Z}_{\geqslant 0}^N$ such that $\blacktriangle_{\boldsymbol{N}}(x) = d$. The dimension of the embedding is the L_1 norm $|x|_1$. Our problem can be stated as an integer linear programming problem

$$\begin{array}{rc} \text{Minimize:} & |x|_1 \\ \text{subject to:} & \blacktriangle_{\boldsymbol{N}}(x) \in Cone(d), \\ & x \in \mathbb{Z}_{\geqslant 0}^N. \end{array}$$

We call a solution to this problem an *optimal embedding*.

Solving integer linear programs is NP-hard. Relaxation of this problem to regular linear programming is not possible since there are no solutions (one can always lower the L_1 norm by going in the direction of the origin).

Solving these kind of problems is closely related to determining their extreme rays. In general, this involves searching an exponentially large space. In Algorithm 2 we show how to find a large enough subset (so that we can get an upper bound on the optimal solution) of the extreme rays in polynomial time.

First we need to prove some nice properties which $\blacktriangle_{\boldsymbol{N}}$ satisfies.

Proposition 11. *Let T be the matrix of $\blacktriangle_{\boldsymbol{N}}$ in the canonical basis of $\mathbb{R}^N$ and T_I be the I-th column ($\varnothing \neq I \subseteq [n]$) of T. Then*

1. $\{T_{ij}\}_{i,j\in[n]} = I_{n\times n}$.
2. *T is symmetric.*
3. $T_{A\triangle B} = T_A \oplus T_B$ *(where $\oplus$ denotes an XOR sum). In particular,* $T_I = \oplus_{i\in I} T_i$.
4. *T has 2^{n-1} ones per row (and column).*
5. $\langle T_I, T_J\rangle = 2^{n-2}$.

Proof. 1) Let $(e^I)_{\varnothing\neq I\subseteq[n]}$ be the canonical basis of R^N. Then,

$$\blacktriangle_{\{j\}}(e^{\{i\}}) = \sum_{l=1}^{|J|}(-2)^{l-1}\sum_{\substack{K\subseteq J\\|K|=l}} \Cap_{\{j\}}(e^{\{i\}}) = \Cap_{\{j\}}(e^{\{i\}}) = \sum_{\varnothing\neq I\subseteq[n]} e_I^{\{i\}}[\{i\}\subseteq I] = [i=j].$$

2) We start by calculating

$$\Cap_K(e^I) = \sum_{J\subseteq[n]} e_K^I[K\subseteq J] = [K\subseteq I].$$

Then,

$$\begin{aligned}\blacktriangle_J(e^I) &= \sum_{l=1}^{|J|}(-2)^{l-1}\sum_{\substack{K\subseteq J\\|K|=l}} \Cap_K(e^I) = \sum_{l=1}^{|J|}(-2)^{l-1}\sum_{|K|=l}[K\subseteq I][K\subseteq J]\\ &= \sum_{l=1}^{|J|}(-2)^{l-1}\sum_{|K|=l}[K\subseteq I\cap J].\end{aligned}$$

Since for $l < |I|$ the sum gives zero it follows that

$$\blacktriangle_J(e^I) = \sum_{l=1}^{\min(|J|,|I|)}(-2)^{l-1}\sum_{|K|=l}[K\subseteq I\cap J] = \blacktriangle_I(e^J).$$

3) The J-wise symmetric difference generalizes the symmetric difference (see Definition 17).

4) This follows directely from Lemma 1.

5) Since T is a binary matrix, $\langle T_I, T_J\rangle$ is the number of coordinates in which both assume the value 1. By item 4 each vector has 2^{n-1} ones. From item 4, $T_{I\triangle J} = T_I \oplus T_J$. Since their XOR also has 2^{n-1} ones, they must coincide in precisely 2^{n-2} entries.

□

This makes it easy to construct the matrix T, by starting with an $n \times n$ identity matrix and then filling the other entries using property 3 of Proposition 11.

We now show how to construct T^{-1}.

Proposition 12. *$T^{-1} = \frac{1}{2^{n-2}} T'$ where $T'_{ij} = [T_{ij} = 1] - [T_{ij} = 0]$.*

Proof. Consider the inner product $\langle T_I, T'_J \rangle$. If $I = J$ the product is the number of ones in T_I, which by Proposition 11 is 2^{n-1}. If $I \neq J$ then the product is the number of ones shared by both T_I and T'_J minus the number of ones that they do not share, which by Proposition 11 (item 5) is zero. □

We are ready to prove our main result.

Algorithm 2. ***Input:*** *a semimetric $d \in \mathbb{R}^N_{>0}$.*

Output: *Approximate optimal embedding for d (an upper bound for $dim_H(d)$).*

1. *Construct T and then T'.*
2. *Find the generators $\{g^1, g^2 \ldots, g^k, \overrightarrow{\mathbf{1}}\}$ of $Cone(d)$.*
3. *For each generator g^i find an extreme ray of $(T' \circ Cone)(d)$, $r^i \in coni(T'(g^i), \overrightarrow{\mathbf{1}})$.*
4. *Output $\sum_{i,j} |r^i_j|$.*

Proof. Step 1: First construct T using Proposition 11 and then T' using Proposition 12.

Step 2: Use Proposition 4.

Step 3: Since $ray(\overrightarrow{\mathbf{1}}) \in int(\mathbb{R}^N_{\geqslant 0})$, there exists an extreme ray $r^i \in coni(T'(g^i), \overrightarrow{\mathbf{1}})$. Thus we need to solve the two dimensional linear optimization problem given by

$$\begin{array}{rc} \text{Minimize:} & |x|_1 \\ \text{subject to:} & x \in coni(T'(g^i), \overrightarrow{\mathbf{1}}), \\ & x \in \mathbb{Z}^N_{\geqslant 0}. \end{array}$$

Since the dimension of this problem is fixed, it is solvable in polynomial time[26][14].

Step 4: Since we have $k+1$ extreme rays inside our $k+1$-dimensional cone, their sum will be in the interior of the cone. Thus, $x = \sum_{i,j} r^i_j$ is an approximate solution for our integer program and $dim_H(d) \leqslant |x|_1$.

□

Since each step is polynomial in N, so is the whole algorithm.

Example 15. *Lets apply Algorithm 2 to* $d = (1, 2, 3, 4, 5, 6, 7)$.

Step 1: Using Propositions 11 and 12 we get

$$T = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \textit{and} \quad T' = \begin{pmatrix} 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 \end{pmatrix}.$$

Step 2: Using proposition 4 and displaying them as a matrix G

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \textit{and} \quad G' = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 0 & 2 & 2 & 0 & 0 & -2 & 0 \\ 1 & 1 & 3 & 1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 0 & 0 & 0 & -2 \\ 1 & 1 & 3 & -1 & 1 & 1 & -1 \\ 0 & 2 & 2 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

where the i*-th row of* G' *is* $T'(g^i)$.

Step 3: For $T'(g^1) = (1, 1, 1, , -1, -1, -1, 1)$, *we want to find* $x \in coni(T'(g^1), \overrightarrow{\mathbf{1}})$ *which minimizes* $|x|_1$. *Thus,* $x = \alpha T'(g^1) + \beta\overrightarrow{\mathbf{1}}$ *with* $\alpha, \beta \geqslant 0$. *This gives us* $x_1 = x_2 = x_3 = x_7$ *and* $x_4 = x_5 = x_6$. *Solving for* α *and* β *we have* $\alpha = \dfrac{x_1 - x_4}{2}$, *from where* $x_4 \leqslant x_1$ *and* $\beta = \dfrac{x_1 + x_4}{2}$. *Thus, the optimal solution takes* $x_1 = 1$ *and* $x_4 = 0$, *and therefore,* $r^1 = (1, 1, 1, 0, 0, 0, 1)$. *Solving for the other* r^i *and displaying them as rows of a matrix* R, *the extreme rays are*

$$R = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Step 4: We get as an approximate solution $x = (4, 6, 8, 1, 1, 2, 1)$ *and thus,* $dim_H(d) \leqslant 23$. *This solution is not optimal since* $(4, 6, 8, 1, 0, 1, 1)$ *is better.*

In the following example the algorithm gives us an optimal solution.

Example 16. *When we apply Algorithm 2 to* $d = (5, 4, 2, 3, 3, 2, 4)$ *we get*

$$R = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

So the approximate solution is $(4, 2, 0, 3, 1, 1, 4)$*. It can be shown (by direct computation) that the algorithm finds all the extreme rays and that these coincide with the Hilbert basis of the cone. Thus the solution is optimal and* $dim_H(d) = 15$*.*

The algorithm also gives optimal solutions when $N = 3$, where it finds all extreme rays and they coincide with the Hilbert basis. An interesting problem would be to characterize when this occurs in general.

As our last result we give general lower and upper bounds on the Hamming cube dimension of a semimetric. We need the two following lemmas.

Lemma 2. *Let* $z \in \mathbb{Z}^N_{\geqslant 0}$*. If* x^* *is a solution to*

$$\begin{array}{ll} \textit{Minimize:} & |x|_1 \\ \textit{subject to:} & x \in \textit{coni}(z, \overrightarrow{\mathbf{1}}), \\ & x \in \mathbb{Z}^N_{\geqslant 0}, \end{array}$$

then $|x^*|_1 \leqslant 2N(\max_{i \in [N]} |z_i|)$*.*

Proof. Since $x^* \in \textit{coni}(z, \overrightarrow{\mathbf{1}})$, there exists $\alpha, \beta \in \mathbb{R}_{\geqslant 0}$ such that $x^* = \alpha z + \beta \overrightarrow{\mathbf{1}}$. Let k be such that $x^*_k = 0$. If such a k does not exist then $x^* - \overrightarrow{\mathbf{1}}$ is also a solution with smaller norm. Thus, $0 = x^*_k = \alpha z_k + \beta$, and therefore $\beta = -\alpha z_k$. So for every i, $x^*_i = \alpha(z_i - z_k)$. Since x^* is optimal, $\alpha \leqslant 1$ (since setting it to 1 gives a solution). Thus, $x^*_i \leqslant (z_i - z_k) \leqslant 2 \max_{i \in [N]} |z_i|$. Summing over all i concludes the proof. □

Lemma 3. *Let* x *be an element of the Hamming cube. Then* $|T'(x)|_\infty \leqslant N$*.*

Proof. Since T' is a $(1, -1)$-matrix, $\langle T_i, x \rangle \leqslant \langle \overrightarrow{\mathbf{1}}, x \rangle \leqslant \langle \overrightarrow{\mathbf{1}}, \overrightarrow{\mathbf{1}} \rangle = N$. □

Theorem 13. *Let* $d \in \mathbb{R}^N_{>0}$ *be such that* $W(d)$ *has* e *equalities. Then*

$$n \leqslant dim_H(d) \leqslant 2N^2(N - e).$$

Proof. The lower bound is the Hamming distance.

Applying Algorithm 2 to d, we find $k = N - e$ extreme rays $(r^1, r^2, \ldots, r^k)$ such that $dim_H(d) \leqslant \sum_i |r^i|_1$. From step 4 of the algorithm each r^i is a solution to an integer programming problem as in Lemma 2. Thus, $|r^i|_1 \leqslant 2N(\max_{i \in [N]} |T'(g^1)|)$. But from Lemma 3, $|T'(g^1)| < N$ from where the result follows. □

6 Future Perspectives

As one could expect, since this thesis introduces new definitions and establish previously unknown relations between channels and coding - basic concepts in the mathematics of communication - and combinatorial structures such as hyperplane arrangements, intersection patterns and geometry of cones, many interesting question and many difficult problems arise from the thesis.

In this chapter we discuss possible research directions for each of the previous chapters.

6.1 Geometry of Communication Channels

In this section we introduced a decoding equivalence between distances. This equivalence is much weaker than the usual scalar equivalence but still interesting. As an example, determining if a distance is isometrically embeddable into the Hamming cube is an NP-hard problem. Yet, as we showed in Theorem 12, under the decoding equivalence every semimetric is embeddable. It would be interesting to study what other properties this equivalence holds.

Another subject we touched in this section is hyperplane arrangements. We showed that the decoding equivalence partitions $\mathbb{R}^n$ into generalized regions of the braid arrangement. In the case of $\mathbb{R}^{n\times m}$, it is partitioned into a product of these. This product is a "deformation" of the braid arrangement, something which is much studied. We firmly believe more results can be obtained by applications of the concepts from the field of hyperplane arrangements.

In section 3.4 we defined a decoding distance on permutations. We showed that it is a weighed version of the Kendall tau rank distance. There are many other distances defined between permutations and it would be interesting to see where our decoding distance fits.

The main question of 3.4 was left open for general dimension decoding cones. Solving this will be equivalent to defining a distance between weak orderings. As in the last paragraph it would be interesting to see how it compares with other distances defined on weak orders.

Finally, with a distance theory for channels, as a long term and ambitious goal, we can construct an approximation theory on them. Say we have a channel P and a family $\mathcal{F}$ of well behaved (in some sense) channels. With a distance we can find which channel from $\mathcal{F}$ best approximates P. As a concrete example, let $\mathcal{F}$ be the family of metrizable

channels. How far away can a channel be from being metrizable? How far a metrizable channel is from a metric determined by a weight (invariant by translations)?

6.2 Channel Metrization

The following enumeration problem is still open: How many channels in Cha_n are metrizable? It is not even clear that this problem is tractable. Many related problems, like counting the number of linear orderings of a general preorder are $\#P$-hard.

We showed how to characterize channel metrization for a single channel. Up to the moment, the unique families that are shown to be metrizable are the symmetric channels ([36]) and the asymmetric channels ([34]). It would be nice to consider interesting families of channels.

6.3 Hamming Cube Embeddings

In Section 5.1 we generalize the notion of intersection patterns to set patterns. We believe that set patterns are interesting mathematical objects to study on their own. Given their generality, it might also be the case that they may be applied to other fields.

Definitions 16 and 17 generalize the notion of intersections and symmetric differences to linear automorphisms of $\mathbb{R}^{2^n-1}$. It would be interesting to explore what other properties of sets could be explored in this geometric way.

It is still not clear how hard it is to determine the Hamming dimension of a distance. It would be interesting to characterize which instances are hard to solve.

Another property of interest is how the packing radius behaves for a metric. In [10] it is shown that determining the packing radius for a code with two words, a triviality with the Hamming metric, is for general metrics an NP-hard problem. In general, the packing radius, $R(C)$, of a linear code, C, for a distance, d, satisfies

$$\left\lfloor \frac{d(C)-1}{2} \right\rfloor \leqslant R(C) \leqslant d(C) - 1.$$

We suspect that a distance with a large packing radius, in terms of the above inequality, will tend to have a large Hamming dimension.

A BIGnav: Bayesian Information Gain for Guiding Multiscale Navigation

The title of this appendix is also the title of a joint work with Wanyu Liu, Michel Beaudouin-Lafon, and Olivier Rioul. This is a very interdisciplinary work, that involved both theoretical aspects (in information theory) and very practical ones (experimental results measuring the efficiency of the human-computer interaction. This work was submitted to an important conference in the area (Conference on Human Factors in Computing Systems - CHI2017) where it was awarded as one of the best papers [27].

In this section we present the theoretical part of the work, the main focus of this author‘s contribution.

A.1 Introduction

Multiscale interfaces are a powerful way to represent large datasets such as maps and deep hierarchies. However, navigating these spaces can be frustrating and inefficient. Most applications, such as Google Maps, only support pan-and-zoom [31]. Others, such as the DragMag [39], use focus+context techniques. In both cases they leave the user in complete control of navigation, leading to frustrating situations such as getting "lost in desert fog" [23].

A few techniques assist navigation by taking advantage of the system's knowledge of the information space: topology-aware navigation [30], visual saliency mediated navigation [22], object pointing [19] and semantic pointing [2] "steer" users towards potential targets, therefore reducing the risk of getting lost.

Other techniques interpret users' intentions to guide navigation: SDAZ [20], for example, adjusts the zoom level according to the user-controlled velocity. While these approaches have proven effective, we believe we can do better by combining them into a more general framework.

We introduce BIGnav, a guided navigation technique that uses both the a priori knowledge of the information space and the progressively acquired knowledge of the user's intention. BIGnav guides navigation through a three-step process:

1. The system interprets user input as an intention revealing what the user is and is not interested in;

2. The system then updates a probabilistic model of the information space to take into account this intention;
3. Finally the system navigates to a new view such that the subsequent user input will maximize the expected information gain of the system.

BIGnav uses Bayesian Experimental Design [29], an approach where the system "runs experiments" on the user to maximize an expected utility. This utility is the information gain, a concept from information theory [37] that represents the amount of information obtained about a variable, here the intended target, from another variable, here the user's input. In other words, BIGnav is a form of human-computer partnership where user and system cooperate to achieve a common objective.

A.2 Background: Bayesian experimental design

Consider a scientist who wants to determine some parameter θ of nature. He can choose to perform an experiment x that will provide an observation y. A probabilistic model is used where Θ, X and Y are the random variables corresponding to θ, x and y, respectively. Bayesian Experimental Design [29] provides a framework to optimize the choice of the experiment x by maximizing an expected utility, commonly defined in terms of the information gained about the parameter θ by the experiment x. The utility may also involve factors such as the financial (or other) cost of performing the experiment.

To optimize the choice of the experiment, the scientist needs two pieces of information, or *priors*:

1. A prior probability distribution $P(\Theta = \theta)$ for all values of θ, which expresses the scientist's knowledge about the random variable Θ before the experiment; and
2. A conditional probability distribution[1] $P(Y = y \mid \Theta = \theta, X = x)$ of the observation Y given the actual value of the parameter θ and the chosen experiment x.

After an experiment x is performed and an observation y is obtained, the scientist updates his knowledge about the parameter θ through Bayes' theorem:

$$P(\Theta = \theta \mid X = x, Y = y) = \frac{P(Y = y \mid \Theta = \theta, X = x)P(\Theta = \theta)}{P(Y = y|X = x)} \tag{1}$$

where $P(Y = y \mid X = x) = \sum_{\theta'} P(Y = y \mid \Theta = \theta', X = x)P(\Theta = \theta')$. This new probability distribution serves as the new prior, on which the scientist can perform another experiment.

[1] The conditional probability $P(A = a \mid B = b)$ reads "the probability of $A = a$ given $B = b$".

The goal of an experiment is to reduce the uncertainty about Θ. As a measure of this uncertainty we use Shannon's entropy function[2] [37]. Initially, the scientist's uncertainty about Θ is given by $H(\Theta)$. After performing an experiment $X = x$ and having observed $Y = y$, the scientist's uncertainty about Θ is given by $H(\Theta|X = x, Y = y)$. The *information gain* is the difference between these two uncertainties:

$$IG(\Theta|X = x, Y = y) = H(\Theta) - H(\Theta|X = x, Y = y). \tag{2}$$

It is generally not possible to know a priori how much information a specific experiment will give[3]. However, for each experiment one can calculate the *expected* information gain[4]:

$$IG(\Theta|X = x, Y) = H(\Theta) - H(\Theta|X = x, Y). \tag{3}$$

To calculate the expected information gain, the scientist uses Bayes' theorem for entropies to convert equation (3) to:

$$IG(\Theta|X = x, Y) = H(Y|X = x) - H(Y|\Theta, X = x) \tag{4}$$

where the first term is given by

$$-\sum_{y} P(Y = y|X = x) \log_2 P(Y = y|X = x)$$

and the second one by

$$-\sum_{y,\theta} P(\Theta = \theta)P(Y = y|\Theta = \theta, X = x) \log_2 P(Y = y|\Theta = \theta, X = x).$$

All the elements in these terms are given by the two priors given to scientist. The scientist can therefore calculate the expected information gain for each possible experiment and choose the experiment that he expects to be most informative.

A.3 BIGnav: Bayesian Information Gain Navigation

The key idea of our approach is to have the system "run experiments" on the user in order to gain information about the user's goal, i.e., the intended target. BIGnav uses Bayesian Experimental Design as follows (Fig. 7a):

[2] The Shannon entropy of a random variable V that takes n possible values, the i-th value of which has probability p_i, is given by:

$$H(V) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Entropy is usually measured in bits and can be interpreted as the level of uncertainty about a variable. It is maximal when all possible values of the variable have the same probability.

[3] Information gain might be negative but is positive on average.

[4] Also known as the mutual information $I(\Theta; Y|X = x)$, which in contrast to equation (2), is always positive.

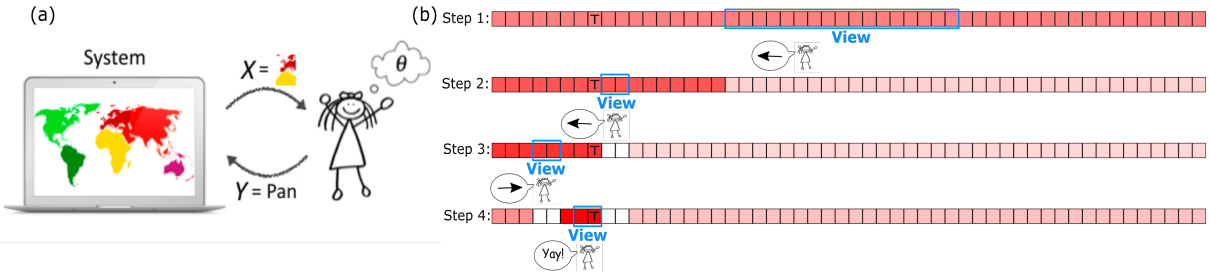

Figure 7 – (a) BIGnav: The system is a scientist experimenting on the user. θ is the intended target in the user's mind. X is the view provided by the system. The user provides an input Y given what she sees in the view (X) and what she wants (θ). (b) Lucy navigates to a particular island (T) among 50 others with BIGnav. The color gradient shows the probability of each island being Lucy's target. The redder, the higher the probability.

- The system plays the role of the scientist;
- The unknown parameter θ is the intended target (known only to the user);
- The experiment x is the view that the system shows to the user after each input; and
- The observation y is the user input after seeing x.

A.3.1 Scenarios

Before describing the process in detail, we illustrate how it works through a series of simple scenarios.

Lucy is a HCI student and is familiar with graphical user interfaces (GUIs). She views a 1D map with some isolated islands and needs to navigate to a particular island.

A.3.1.1 Scenario 1

There are two islands on the map. Lucy is currently on island A and needs to navigate to the other island, B, which she knows is to her left. Instead of zooming out until island B appears in the view and then zooming in, or using a long series of pans to the left over the ocean, Lucy uses BIGnav and gives a left command. BIGnav interprets this action as "Lucy wants something else other than island A", and updates its probability distribution. Since there are only two islands on the map, BIGnav directly shows island B as if to ask: "Is this what you are looking for?" Lucy happily clicks on it.

A.3.1.2 Scenario 2

There are three islands on the map. Lucy needs to navigate from island A to island B to her left, but island C is in between. She uses BIGnav again and gives it a left

command. If BIGnav does not know anything about Lucy's intention, it will show island C in the middle and wait for Lucy's confirmation. Then Lucy pans to the left again, BIGnav locates island B since both islands A and C are excluded. However, if there is available prior knowledge, e.g., that 90% of the people visit island B instead of island C, BIGnav will show island B directly.

If islands B and C are on opposite sides of island A, Lucy's first action would directly determine what she wants, for instance, going to the left would lead directly to island B. However, if Lucy makes a mistake and gives the wrong direction, the system would show her island C. She realizes that this is not what she is looking for, she issues another command to the correct direction, the system then takes her directly to island B since BIGnav dynamically updates its knowledge about Lucy's interest.

A.3.1.3 Scenario 3

Lucy now has the difficult task of navigating from island A to one of the 50 islands on a 1D map. At each step she can go left, go right or zoom in. BIGnav interprets each command, updates its knowledge and shows Lucy a view where her next command is most likely to maximize the reduction of uncertainty (or information gain) about the intended target. Figure 7(b) shows a similar scenario where BIGnav guides Lucy to her target in 4 steps.

A.3.2 Detailed Description

We now describe in detail how BIGnav uses Bayesian experimental design and information gain to guide navigation. The three key random variables are:

- Θ represents any point of interest, i.e., possible intended target. For each target θ, and the probability that it is the actual intended target is $P(\Theta = \theta)$. These probabilities constitute the a priori knowledge that the system has about the user's interest, and is updated as the user navigates.
- X represents any possible view provided by the system. $X = x$ is a particular view shown to the user. Note that the number of possible views is potentially very large.
- Y represents any particular command y issued by the user. The possible input commands are: move towards a direction, zoom in or click on the target when it is big enough to be clickable. Note that zooming out is not required in this framework: if the target is out of view, the user should indicate in which direction it is rather than zooming out.

We now describe the three-stage navigation process.

(1) Interpreting user input: Given the view x shown to the user and the user's intended target θ, $P(Y = y|\Theta = \theta, X = x)$ is the probability that the user provides

an input command $Y = y$ given θ and x. This probability distribution is the system's interpretation of the user's intention when giving this command. For instance, if island B is to the left of Lucy, what is the probability of Lucy giving the left command when knowing that island B is located to her left? $P(\textit{go left} \mid \textit{island B is the intended target, island B is located to the left of the current view}) = 1$ if Lucy is completely confident about what she is doing. But maybe Lucy is not accurate all the time. Say she is only correct 95% of time, then we need to consider that she makes errors. For instance, $P(\textit{go left} \mid \textit{island B is the intended target, island B is located to the left of the current view}) = 0.95$ and $P(\textit{go right} \mid \textit{island B is the intended target, island B is located to the left of the current view}) = 0.05$. $P(Y|\Theta = \theta, X = x)$ is a priori knowledge that must be given to the system.

(2) Updating system's knowledge: Given the view x shown to the user and the user reaction y to that view, the system can update its estimate $P(\Theta|X = x, Y = y)$ of the user's interest with equation (1). If the system has no prior knowledge about the user's intended target, e.g., at the beginning, each θ has the same probability of being the target and $P(\Theta)$ is uniform. As the user issues commands, the system gains knowledge about the likelihood that each point of interest be the target, reflected by the changes to the probability distribution. This is done, for each point of interest, by taking its previous probability, multiplying by the above user input function $P(Y = y|\Theta = \theta, X = x)$, and normalizing it so that the sum of the new probabilities over all the points of interest equals one.

(3) Navigating to a new view: With the new probability distribution after receiving user input, BIGnav then goes over each view $x \in X$, calculates its expected information gain with equation (4) and picks the view for which it is maximal. To maximize equation (4), BIGnav looks for a trade-off between two entropies. To maximize the first term, the view should be such that all user commands given that view are equally probable (for the system). To minimize the second term, the view should provide the user with meaningful information about the points of interest. Maximizing a difference does not necessarily mean to maximize the first term and minimize the second, so the maximum information gain is a trade-off between these two goals. For example, showing only ocean will increase the first term but will also increase the second term. After locating the view with maximal information gain, BIGnav navigates there and waits for user's next input.

A.4 BIGnav in 1D

The 50 islands are the points of interest, therefore $\Theta = \{1, 2, \ldots, 50\}$. The system does not have prior knowledge about Lucy's intended target island, so the initial distribution is $P(\Theta_1 = i) = \dfrac{1}{50}$.

The view presented to Lucy at each step is defined by $X = \{[a, b] \subseteq [1, 50]\}$.

The maximum zoom factor is such that a view cannot be smaller than two blocks ($b - a \leqslant 2$). Since it is a 1D map, Lucy can go to the left, go to the right, zoom in or select the target if the view is at the full scale. We note these commands $Y = \{\leftarrow, \rightarrow, + \text{ (zoom in)}, \bullet \text{ (click target i)}\}$.

We start by modeling Lucy's behavior. We consider that Lucy makes some mistakes when panning and zooming, but will not miss the target when it is shown in the view and clickable:

$$P(Y = \rightarrow \mid \Theta = \theta, X = [a,b]) = \begin{cases} 0.9 & b < \theta \\ 0.05 & a < \theta \\ 0.05 & a \leqslant \theta \leqslant b \text{ and } b - a > 2 \\ 0 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \end{cases}$$

$$P(Y = \leftarrow \mid \Theta = \theta, X = [a,b]) = \begin{cases} 0.05 & b < \theta \\ 0.9 & a < \theta \\ 0.05 & a \leqslant \theta \leqslant b \text{ and } b - a > 2 \\ 0 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \end{cases}$$

$$P(Y = + \mid \Theta = \theta, X = [a,b]) = \begin{cases} 0.05 & b < \theta \\ 0.05 & a < \theta \\ 0.9 & a \leqslant \theta \leqslant b \text{ and } b - a > 2 \\ 0 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \end{cases}$$

$$P(Y = \bullet \mid \Theta = \theta, X = [a,b]) = \begin{cases} 1 & a \leqslant i = \theta \leqslant b \text{ and } b - a \leqslant 2 \\ 0 & \text{otherwise.} \end{cases}$$

In Fig. 7b, the islands are represented by square boxes and colored in shades of red indicating the degrees to which the system believes the island is the target, i.e., island i is darker than j if $P(\Theta = i) > P(\Theta = j)$. Island 8 has a **T** indicating that it is the target. The blue rectangle is the view that the system shows to Lucy. After seeing the view, Lucy provides an input command y to the system.

We can now show BIGnav in action.

Step 1: Since the initial distribution is uniform, the system's uncertainty about Lucy's target is $H_1 = H(\Theta_1) = \log_2 50 = 5.64$ bits.

The system then goes over every image $[a,b]$, finds that $[18,34]$ maximizes the expected information gain and displays the corresponding initial view to Lucy. In this case the expected information gain from Lucy's next action is $IG(\Theta_1 \mid X = [18,34], Y) = 1.08$ bits.

Lucy inputs $\leftarrow$ after seeing $[18,34]$. The system then updates its knowledge with equation (1) and ends up with a new distribution Θ_2 given by $P(\Theta_2) = P(\Theta_1 \mid X =$

$[18, 34], Y = \leftarrow$). Using Bayes' theorem we have:

$$P(\Theta_2 = i) = \begin{cases} 0.05 & i < 18 \\ 0.002 & i \geqslant 18. \end{cases}$$

The updated uncertainty is $H_2 = H(\Theta_2) = 4.65$ bits, resulting in an actual information gain $H_1 - H_2 = 0.99$ bits, very close to the expected information gain of 1.08 bits.

Step 2: The system now searches for the best view using the new distribution $P(\Theta_2)$, finds that it is $[9, 10]$ with an expected information gain of $IG(\Theta_2 \mid X = [9, 10], Y) = 1.24$ bits and displays it to Lucy. She then inputs $\leftarrow$ after seeing $[9, 10]$. The system then updates Θ_2 to Θ_3 as follows:

$$P(\Theta_3 = i) = \begin{cases} 0.12 & i < 9 \\ 0 & 9 \leqslant i \leqslant 10 \\ 0.006 & 10 < i < 18 \\ 0.0003 & i \geqslant 18. \end{cases}$$

The entropy of Θ_3 is $H_3 = 3.36$ bits, so the actual information gain for this step is $H_2 - H_3 = 1.29$ bits, higher than the expected information gain of 1.24 bits.

Step 3: With the same process, the best view is now $[4, 5]$ with an expected information gain of $IG(\Theta_3 \mid X = [4, 5], Y) = 1.58$ bits. Lucy inputs $\rightarrow$, leading to the updated distribution

$$P(\Theta_4 = i) = \begin{cases} 0.01 & i < 4 \\ 0 & 4 \leqslant i \leqslant 5 \\ 0.28 & 5 < i < 9 \\ 0 & 9 \leqslant i \leqslant 10 \\ 0.015 & 10 < i < 18 \\ 0.0007 & i \geqslant 18. \end{cases}$$

The entropy of Θ_4 is $H_4 = 2.70$ bits, so the actual information gain is $H_3 - H_4 = 0.66$ bits, compared to the expected information gain of 1.58 bits.

Step 4: The best view is now $[7, 8]$ with an expected information gain of $IG(\Theta_4 \mid X = [7, 8], Y) = 1.84$ bits. Lucy sees that the target island is in the view and happily clicks on it. The updated distribution is updated to

$$P(\Theta_5 = i) = \begin{cases} 1 & i = 8 \\ 0 & \text{otherwise.} \end{cases}$$

The entropy of Θ_5 is $H_5 = 0$ bits since there is no more uncertainty about the target, so the actual information gain is $H_4 - H_5 = 2.7$ bits, while the expected information gain was 1.84 bits.

Lucy finds her target island in only 4 steps. At step 1, BIGnav divides the map in 3 so that the three commands (left, right and zoom in) have equal probability. It

does not consider a click as the view is still far from being fully zoomed-in. At step 2, one would expect it to divide the left third of the map in 3 again so that the view would be about 5 boxes wide. However, since it is close to a fully zoomed-in scale, and it knows that Lucy never misses her target when it is in the view and is clickable, showing a 2-box zoomed-in view will give BIGnav extra information: if this is the target, Lucy will click on it; if it is not and Lucy moves away, the probabilities of these two boxes become 0. Step 3 and step 4 work similarly.

We ran 200 simulations with 50 islands and a uniform initial distribution and found that it required 3.3 steps on average.

A.5 Conclusion and future work

BIGnav is a new multiscale navigation technique based on Bayesian Experimental Design with the criterion of maximizing the information-theoretic concept of mutual information. At each navigation step, BIGnav interprets user input, updates its estimate of the user's intention, and navigates to a view that maximizes the expected information that will be gained from the user's subsequent input.

We ran a controlled experiment (details can be found in [27]) comparing BIGnav with standard pan and zoom for different levels of difficulty and different distributions of the information space. Our main result is that BIGnav is up to 40.0% faster than the baseline for distant targets and non-uniform information spaces.

To the best of our knowledge, BIGnav is the first attempt at introducing an information-theoretic and Bayesian approach to multiscale navigation. Our next goal is to reduce users' cognitive load while still ensuring BIGnav's efficiency. We also want to improve the computational cost of the technique in order to support more input commands and a finer grid.

Beyond navigation, we are interested in applying this approach to other tasks, such as searching. Indeed, the model can be framed in terms of human-computer interaction as follows:

- X can be any system feedback, e.g., visual, auditory, haptic;
- Y can be any human input, e.g., touch input or gaze;
- $P(\Theta)$ can model many kinds of prior knowledge about users' goals, as well as reflect their interaction history.

The paradigm shifts from "responding to user input" to "running experiments on the user" is a novel perspective on the notion of human-computer partnerships, and the Bayesian Information Gain model opens the door to a wide range of "BIG" applications.

Bibliography

[1] D. Burago, Y. Burago and S. Ivanov, "A Course in Metric Geometry," American Mathematical Society, 2001.

[2] R. Blanch, Y. Guiard and M. Beaudouin-Lafon,"Semantic Pointing: Improving Target Acquisition with Control-display Ratio Adaptation," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 519–526, ACM, 2004.

[3] C. Carlet, "Z2K-linear Codes", *IEEE Transactions on Information Theory*,Volume: 44, Issue: 4, p. 1543–1547, 1998 .

[4] V. Chvátal, "Recognizing intersection patterns," Ann.Discrete Math. 8, p. 249–251, 1980.

[5] T. Cover and J.A. Thomas, "Elements of information theory,"2nd ed., *New York: Wiley-Interscience.*, 1980.

[6] M. Deza, "Matrices de formes quadratiques non negatives pour des arguments binaires," C.R. Acad. Sc. Paris, vol. 277, p. 873–875, 1973.

[7] M. Deza and M. Laurent, "Geometry of cuts and metrics," Algorithms and Combinatorics 15, *Springer*, 1997.

[8] M. Deza and E. Deza, "Encyclopedia of distances," 4th revised edition, *Springer-Verlag*, 2016.

[9] E. Deza, M. Deza and M.S. Dutour, "Generalizations of Finite Metrics and Cuts," *World Scientific*, 2016.

[10] R.G.L. D'Oliveira and M. Firer, "The packing radius of a code and partitioning problems: The case for poset metrics on finite vector spaces," , *Discrete Mathematics*, Vol. 338, Issue 12, p. 2143–2167, 2015.

[11] R.G.L. D'Oliveira and M. Firer, "Embedding Distances into the Hamming Cube," *Proceedings of the 9th International Workshop on Coding and Cryptography*, 2015.

[12] R.G.L. D'Oliveira and M. Firer, "Geometry of Communication Channels: Metrization and Decoding," *Symmetry: Culture and Science*, Volume 27, No. 4, p. 279–289, 2016.

[13] R.G.L. D'Oliveira and M. Firer, "Minimum Dimensional Hamming Cube Embeddings," to appear in *Advances in Mathematics of Communications*, 2017.

[14] F. Eisenbrand, "Fast integer programming in fixed dimension," *11th Annual European Symposium Proceedings*, Springer Berlin Heidelberg, p. 196–207, 2003.

[15] M. Firer and J.L. Walker, "Matched Metrics and Channels", *IEEE Transactions on Information Theory*,Volume: 62, Issue: 3, p. 1150–1156, 2015.

[16] E. Gabidulin, "A brief survey of metrics in coding theory," *Mathematics of Distances and Applications*, p. 66–84, 2012.

[17] M. Greferath and S.E. Schmidt, "Gray Isometries for Finite Chain Rings and a Nonlinear Ternary (36, 312, 15) Code", *IEEE Transactions on Information Theory*, Volume: 45, Issue: 7, p. 2522–2524, 1999.

[18] P. M. Gruber, "Convex and discrete geometry," *Springer-Verlag, New York*, 2007.

[19] Y. Guiard, R. Blanch and M. Beaudouin-Lafon, "Object Pointing: A Complement to Bitmap Pointing in GUIs," *Proceedings of Graphics Interface 2004*, p. 9–16, Canadian Human-Computer Communications Society, 2004.

[20] T. Igarashi and K. Hinckley, "Speed-dependent Automatic Zooming for Browsing Large Documents," *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, p. 139–148, ACM, 2000.

[21] K.E. Iverson, "A Programming Language," *John Wiley & Sons*, 1962.

[22] W. Javed, S. Ghani and N. Elmqvist, "GravNav: Using a Gravity Model for Multiscale Navigation," *Proceedings of the International Working Conference on Advanced Visual Interfaces*, p. 217–224, ACM, 2012.

[23] S. Jul and G.W. Furnas, "Critical Zones in Desert Fog: Aids to Multiscale Navigation," *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, p. 97–106, ACM, 1998.

[24] M. Kendall, "A New Measure of Rank Correlation," *Biometrika*, 30, p. 81–89, 1938.

[25] D.E. Knuth, "Two notes on notation", *Am. Math. Monthly 99*, p. 403–422, 1992.

[26] H.W. Lenstra Jr, "Integer programming with a fixed number of variables", *Mathematics of operations research 8.4*, p. 538–548, 1983.

[27] W. Liu, R.G.L. D'Oliveira, M. Beaudouin-Lafon, and O. Rioul, "BIGnav: Bayesian Information Gain for Guiding Multiscale Navigation", to appear in *CHI Conference on Human Factors in Computing Systems*, 2017.

[28] J.L. Massey, "Notes on coding theory, class notes for course 6.575 (spring)", M.I.T., Cambridge, MAA, 1967.

[29] D.V. Lindley, “On a measure of the information provided by an experiment”, *The Annals of Mathematical Statistics*, p. 986–1005, JSTOR, 1956.

[30] T. Moscovich, F. Chevalier, N. Henry, E. Pietriga and J. Fekete, “Topology-aware Navigation in Large Networks”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 2319–2328, ACM, 2009.

[31] Perlin, K. and Fox, D. “Pad: An Alternative Approach to the Computer Interface", on *Proc. of the 20th Annual Conf. on Computer Graph. and Int. Tech. - SIGGRAPH '93*, p. 57–64, ACM, 1993.

[32] V. Pless, “Introduction to the Theory of Error-Correcting Codes”, *Wiley-Interscience Series in Discrete Mathematics*, 1982.

[33] A. Poplawski, “On Matched Metric and Channel Problem”, arXiv:1606.02763 [cs.IT], 2016.

[34] C. Qureshi, “Matched Metrics to the Binary Asymmetric Channels”, arXiv:1606.09494 [cs.IT], 2016.

[35] B.S.W. Schroder, “Ordered Sets: An Introduction,” *Boston: Birkhäuser*, 2002.

[36] G. Séguin, “On metrics matched to the discrete memoryless channel”, *J. Franklin Inst. 309*, no. 3, p. 179–189, 1980.

[37] C.E. Shannon, “ A Mathematical Theory of Communication”. *Bell System Technical Journal. 27*, p. 379–423, 1948.

[38] R.P. Stanley, “ An Introduction to Hyperplane Arrangements”. *Lecture notes, IAS/-Park City Mathematics Institute*, 2004.

[39] C. Ware and M. Lewis, “ The DragMag Image Magnifier”. *Conference Companion on Human Factors in Computing Systems*, p. 407–408, ACM, 1995.