Technical University of Catalonia (UPC)
Department of Signal Theory and Communications (TSC)

Graph-based Techniques for Compression and Reconstruction of Sparse Sources

Ph.D. dissertation by
Francisco Ramírez-Jávega
Advisor
Meritxell Lamarca Orozco

Barcelona, November 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de qualificació de tesi doctoral

Curs acadèmic:

Nom i cognoms

Programa de doctorat

Unitat estructural responsable del programa

Resolució del Tribunal

Reunit el Tribunal designat a l'efecte, el doctorand / la doctoranda exposa el tema de la seva tesi doctoral titulada

_____

_____.

Acabada la lectura i després de donar resposta a les qüestions formulades pels membres titulars del tribunal, aquest atorga la qualificació:

☐ NO APTE ☐ APROVAT ☐ NOTABLE ☐ EXCEL·LENT

(Nom, cognoms i signatura) President/a	(Nom, cognoms i signatura) Secretari/ària	
(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal

_____, _____ d'/de _____ de _____

El resultat de l'escrutini dels vots emesos pels membres titulars del tribunal, efectuat per l'Escola de Doctorat, a instància de la Comissió de Doctorat de la UPC, atorga la MENCIÓ CUM LAUDE:

☐ SÍ ☐ NO

(Nom, cognoms i signatura) President de la Comissió Permanent de l'Escola de Doctorat	(Nom, cognoms i signatura) Secretari de la Comissió Permanent de l'Escola de Doctorat

Barcelona, _____ d'/de _____ de _____

Abstract

The main goal of this thesis is to develop lossless compression schemes for analog and binary sources. All the considered compression schemes have as common feature that the encoder can be represented by a graph, so they can be studied employing tools from modern coding theory.

In particular, this thesis is focused on two compression problems: the group testing and the noiseless compressed sensing problems. Although both problems may seem unrelated, in the thesis they are shown to be very close. Furthermore, group testing has the same mathematical formulation as non-linear binary source compression schemes that use the OR operator. In this thesis, the similarities between these problems are exploited.

The group testing problem is aimed at identifying the defective subjects of a population with as few tests as possible. Group testing schemes can be divided into two groups: adaptive and non-adaptive group testing schemes. The former schemes generate tests sequentially and exploit the partial decoding results to attempt to reduce the overall number of tests required to label all members of the population, whereas non-adaptive schemes perform all the test in parallel and attempt to label as many subjects as possible.

Our contributions to the group testing problem are both theoretical and practical. We propose a novel adaptive scheme aimed to efficiently perform the testing process. Furthermore, we develop tools to predict the performance of both adaptive and non-adaptive schemes when the number of subjects to be tested is large. These tools allow to characterize the performance of adaptive and non-adaptive group testing schemes without simulating them.

The goal of the noiseless compressed sensing problem is to retrieve a signal from its lineal projection version in a lower-dimensional space. This can be done only whenever the amount of null components of the original signal is large enough. Compressed sensing deals with the design of sampling schemes and reconstruction algorithms that manage to reconstruct the original signal vector with as few samples as possible.

In this thesis we pose the compressed sensing problem within a probabilistic framework, as opposed to the classical compression sensing formulation. Recent results in the state of the art show that this approach is more efficient than the classical one.

Our contributions to noiseless compressed sensing are both theoretical and practical. We deduce a necessary and sufficient matrix design condition to guarantee that the reconstruction is lossless. Regarding the design of practical schemes, we propose two novel reconstruction algorithms based on message passing over the sparse representation of the matrix, one of them with very low computational complexity.

Keywords: "Noiseless compressed sensing", "Group testing", "Adaptive group testing", "Verification algorithm", "Analog compression", "sparse pattern recovery","Binary source compression", "lossless reconstruction"

Resumen

El objetivo principal de la tesis es el desarrollo de esquemas de compresión sin pérdidas para fuentes analógicas y binarias. Los esquemas analizados tienen en común la representación del compresor mediante un grafo; esto ha permitido emplear en su estudio las herramientas de codificación modernas.

Más concretamente la tesis estudia dos problemas de compresión en particular: el diseño de experimentos de testeo comprimido de poblaciones (de sangre, de presencia de elementos contaminantes, secuenciado de ADN, etcétera) y el muestreo comprimido de señales reales en ausencia de ruido. A pesar de que a primera vista parezcan problemas totalmente diferentes, en la tesis mostramos que están muy relacionados. Adicionalmente, el problema de testeo comprimido de poblaciones tiene una formulación matemática idéntica a los códigos de compresión binarios no lineales basados en puertas OR. En la tesis se explotan las similitudes entre todos estos problemas.

Existen dos aproximaciones al testeo de poblaciones: el testeo adaptativo y el no adaptativo. El primero realiza los test de forma secuencial y explota los resultados parciales de estos para intentar reducir el número total de test necesarios, mientras que el segundo hace todos los test en bloque e intenta extraer el máximo de datos posibles de los test.

Nuestras contribuciones al problema de testeo comprimido han sido tanto teóricas como prácticas. Hemos propuesto un nuevo esquema adaptativo para realizar eficientemente el proceso de testeo. Además hemos desarrollado herramientas que permiten predecir el comportamiento tanto de los esquemas adaptativos como de los esquemas no adaptativos cuando el número de sujetos a testear es elevado. Estas herramientas permiten anticipar las prestaciones de los esquemas de testeo sin necesidad de simularlos.

El objetivo del muestreo comprimido es recuperar una señal a partir de su proyección lineal en un espacio de menor dimensión. Esto sólo es posible si se asume que la señal original tiene muchas componentes que son cero. El problema versa sobre el diseño de matrices y algoritmos de reconstrucción que permitan implementar esquemas de muestreo y reconstrucción con un número mínimo de muestras. A diferencia de la formulación clásica de muestreo comprimido, en esta tesis se ha empleado un modelado probabilístico de la señal. Referencias recientes en la literatura demuestran que este enfoque permite conseguir esquemas de compresión y descompresión más eficientes.

Nuestras contribuciones en el campo de muestreo comprimido de fuentes analógicas dispersas han sido también teóricas y prácticas. Por un lado, la deducción de la condición necesaria y suficiente que debe garantizar la matriz de muestreo para garantizar que se puede reconstruir

unívocamente la secuencia de fuente. Por otro lado, hemos propuesto dos algoritmos, uno de ellos de baja complejidad computacional, que permiten reconstruir la señal original basados en paso de mensajes entre los nodos de la representación gráfica de la matriz de proyección.

Palabras clave: "Noiseless compressed sensing", "Group testing", "Adaptive group testing", "Verification algorithm", "Analog compression", "sparse pattern recovery"

Acknowledgements

I am deeply indebted to my advisor Prof. Meritxell Lamarca for her advice and support, without which this dissertation would be impossible. She taught me how to formulate a problem in the cleanest possible way. In addition, I appreciate his endless patience and the freedom she gave to me, allowing me to pursue my research interests that sometimes were different from hers. I am also grateful to Prof. Francesc Rey for being my first and second-year academic advisor. I would also like to acknowledge Prof. Gregori Vazquez, Prof. Javier Villares, Prof. Josep Sala and Prof. Jaume Riba for stimulating discussions.

I have been fortunate to have many friends and colleagues who made these last years an unforgettable experience. In particular, I thank David Matas for numerous rewarding discussions, during every one of which I gained new knowledge. And of course to dear friends: Maruan, Jokin, Pasca, Oscar, Mireia, Juan, Javi and Montse for being always there for me during both joyful and stressful times, to whom I will always be indebted.

Finally, it is my greatest honor to thank my family: my parents Francisco and Maria for their support, self-sacrifice, never-ending wisdom, unbounded freedom and unconditional love they have given to me since I was born. They are the ones responsible of my inherent curiosity and inherited lateral thinking. My brothers Miguel and Joan for being there when I needed them and, finally, Matilde, my niece who is just taking her first steps and reminds me what really matters.

Contents

List of Figures

List of Tables

Acronyms

Notation	Description
AC	analog compression
AMP	approximate message passing
BEC	binary erasure channel
BER	bit error rate
BP	belief propagation
BSC	binary symmetric channel
COMP	combinatorial optimal matching pursuit
CoSaMP	compressed sampling matching pursuit
CS	compressed sensing
CS-LMP	list message passing for compressed sensing
DD	definite defective
EM-GB-AMP	estimate-maximize Gaussian-Bernoulli approximate message passing
EVA	enhanced verification algorithm
FFT	fast fourier transform
GAMP	generalized approximate message passing
GT	group testing
IST	iterative soft thresholding

Notation	Description
ITH	iterative hard thresholding
LASSO	least absolute shrinkage and selection operator
LDPC	low density parity-check
LDPrC	low density product-check
LMP	list message passing
MAP	maximum a posteriori
MMSE	minimum mean square error
MP	matching pursuit
NB-VA	node-based verification algorithm
OMP	orthogonal matching pursuit
RIP	restricted isometry property
ROMP	regular orthogonal matching pursuit
SAMP	sparsity adaptive matching pursuit
SER	sequence error rate
SuPrEm	sum-product expectation maximization
tuned TST	tuned two-stage thresholding
VA	verification algorithms
WER	word error rate

List of Symbols

Notation	Description
$\mathcal{G}$	Graph
δ	Efficiency of a compression scheme
δ^{BP}	Maximum compression efficience of an ensemble of graphs in N-asymptotic regime
Γ	Ensemble of graphs
γ	Sparsity rate of a source
γ^{BP}	Belief propagation threshold of a ensemble of graphs in the N-asymptotic regime
$\bar{d}_c^i$	Average number of edges per check node of the graph $\mathbf{A}_i$.
d_c	Number of edges of a check node
$\bar{d}_v^i$	Average number of edges per variable node of the graph $\mathbf{A}_i$.
d_v	Number of edges of a variable node
A(i)	Succession of matrices/actions generated upto instant i
K	Amount of non-null components of vector $\mathbf{x}_0$
L$_i(d)$	Variable node degree distribution of graph $\mathbf{A}_i$
L(d)	Variable node degree distribution of graph $\mathbf{A}$
M	Number of components of vector $\mathbf{y}_0$
N	Number of components of vector $\mathbf{x}_0$

Notation	Description
$\mathrm{R}_i(d)$	Check node degree distribution of graph $\mathbf{A}_i$
$\mathrm{R}(d)$	Check node degree distribution of graph $\mathbf{A}$
r_i	Rate of a matrix $\mathbf{A}_i$
$\mathbf{A}_i$	Sub-matrix of $\mathbf{A}$ generated at instant i
$\mathbf{A}$	Measurement matrix
$\mathbf{e}$	Noise or error sequence
$\mathbf{x}_0$	Source sequence
$\mathbf{x}$	Sequence
$\mathrm{y}(i)$	Succession of observations generated upto instant i
$\mathbf{y}_i$	Vector of observations generated at instant i
$\mathbf{y}$	Sequence at output channel
$\Delta\mathcal{D}_i$	Set of defective subjects labeled at instant i but unlabeled at instant $i-1$.
$\Delta\mathcal{D}_m^i$	Set of defective subjects labeled by $\mathrm{g}(\mathbf{A}_m,\mathbf{y}_m)$ at instant j that were unlabeled at instant $j-1$.
$\Delta\mathcal{N}_i$	Set of non-defective subjects labeled at instant i but unlabeled at instant $i-1$.
$\Delta\mathcal{N}_m^i$	Set of non-defective subjects labeled by $\mathrm{g}(\mathbf{A}_m,\mathbf{y}_m)$ at instant j that were unlabeled at instant $j-1$.
$\Gamma_{\mathrm{N}}^{\mathrm{K}}$	Grassmannian of K-dimensional sub-spaces of the N-dimensional space $\mathbb{R}^{\mathrm{N}}$
$\mathcal{A}_u$	Null-space of matrix $\mathbf{A}$
$\mathcal{D}$	Set of defective subjects labeled at the end of the decoding process
$\mathcal{D}^c$	Set of defective subjects that remain unlabeled at the end of the decoding process

Notation	Description
$\mathcal{D}_i$	Set of defective subjects of the population labeled up-to i-th stage
$\mathcal{D}_i^c$	Set of defective subjects of the population that remain unlabeled at the end of the i-th stage
$\mathcal{D}_{min}$	Set of defective subjects of the population that can be retrieved from a matrix
$\mathcal{D}_{min}^c$	Set of defective subjects of the population that cannot be retrieved from a matrix
$\mathcal{D}_m^i$	Set of defective subjects labeled by g($\mathbf{A}_m$,$\mathbf{y}_m$) at instant j.
$\mathcal{J}_i^c$	Neighborhood of check node i.
$\mathcal{J}_i^v$	Neighborhood of variable node i.
$\mathcal{N}$	Set of non-defective subjects labeled at the end of the decoding process
$\mathcal{N}^c$	Set of non-defective subjects that remain unlabeled at the end of the decoding process
$\mathcal{N}_i$	Set of non-defective subjects of the population labeled up-to i-th stage
$\mathcal{N}_i^c$	Set of non-defective subjects of the population that remain unlabeled at the end of the i-th stage
$\mathcal{N}_{min}$	Set of non-defective subjects of the population that can be retrieved from a matrix
$\mathcal{N}_{min}^c$	Set of non-defective subjects of the population that cannot be retrieved from a matrix
$\mathcal{N}_m^i$	Set of non-defective subjects labeled by g($\mathbf{A}_m$,$\mathbf{y}_m$) at instant j
$\mathcal{N}_y^i$	Set of non-defective tests generated at instant i.
$\mathcal{P}$	Population set
$\mathcal{P}_D$	Set of defective subjects of the Population
$\mathcal{P}_N$	Set of non-defective subjects of the Population

Notation	Description
$\mathcal{S}$	Support set
$\mathcal{U}$	Set of subjects that remain unlabeled at the end of the decoding process
$\mathcal{U}^c$	Set of subjects that labeled at the end of the decoding process
$\mathcal{U}_i$	Set of subjects of the population that remain unlabeled at the end of the i-th stage
$\mathcal{U}_i^c$	Set of subjects of the population labeled up-to i-th stage
$\mathcal{U}_{min}$	Set of subjects of the population that cannot be retrieved from a matrix
$\mathcal{V}_i$	The i-th K-dimensional sub-space of the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$
$\mathcal{V}_i^e$	Set of erroneous sequences that belong to to partition $\mathcal{V}_i$
$\mathcal{V}_{i\leftarrow j}^e$	Set of all the sequences in $\mathcal{V}_i^e$ that have a pair sequence in $\mathcal{V}_j$
$\mathcal{X}$	Domain of the encoder
$\mathcal{X}^b$	Sub-space of the domain of a encoder that encloses the pints where the encoder is injective
$\mathcal{X}^e$	Sub-space of the domain of a encoder that encloses the pints where the encoder is not injective
$\mathcal{Y}$	Codomain of the encoder
$\mathcal{Y}^b$	Sub-space of the codomain of a encoder function that encloses all the points where the encoder is injective
$\mathcal{Y}^e$	Sub-space of the codomain of a encoder function that encloses all the points where the encoder is not injective

Glossary

Notation	Description
$\lvert\cdot\rvert$	Counting measure
$\lVert\cdot\rVert_0$	Pseudo-norm zero, l_0
$\setminus$	Excluding
$\mathrm{d}(X)$	Renyi dimension of the random variable X
$\bar{\mathrm{d}}(X)$	Upper Renyi dimension of the random variable X
$\underline{\mathrm{d}}(X)$	Lower Renyi dimension of the random variable X
$\mathrm{f}(X)$	Probability density function or X
$\mathrm{g}(\cdot)$	Decoding function
$\mathrm{Lip}(\cdot)$	Lipschitz pseudo-norm
$\mathrm{O}(x)$	Big O notation
$\mathrm{o}(x)$	Small O notation
$\mathrm{P}(X)$	Probability mass function X
$\mathrm{q}(\cdot)$	Encoding function
$\mathrm{q}_a(\cdot)$	AND-based encoder
$\mathrm{q}_o(\cdot)$	OR-based encoder
$\mathrm{q}_x(\cdot)$	XOR-based encoder
$\mathrm{spark}(\cdot)$	Minimum number of linear dependent vectors of a matrix

Notation	Description
$\mu_{\mathrm{K}}(\cdot)$	K-dimensional Lebesgue measure

PART I

Problem description and theoretical limits

CHAPTER 1

Sensing structured sources

This thesis proposes contributions on several topics that can be formulated under the common analog compression (AC) framework [161]: binary source compression [75, 170, 171], group testing [51], sparse pattern recovery [59, 108, 110] and compressed sensing (CS) [28, 43].

The AC framework was recently introduced by Wu [161] and deals with the compression of signals generated by sources that can be under-sampled and reconstructed with an arbitrary distortion.

This thesis is divided in three blocks. Part I is devoted to review the state of the art. First, in this chapter we describe the analog compression framework and review briefly the CS, the source compression, the group testing and the sparse pattern recovery problems, stressing the existing relationship between them. Afterwards, chapters 2 and 3 are devoted to review the state of the art in the binary source compression and compressed sensing schemes respectively.

Part II of the thesis focuses on binary source compression. There we present our findings in the study of the behaviour of a non-linear binary graph-based source compression encoder where each component of the compressed sequence is generated by the OR-logical operation of several components of the original sequence. This simple encoder is also employed to study the group testing and sparse pattern recovery problems.

In chapter 4 we characterize the encoding limits of a non-adaptive group testing encoder, the performance of a non-adaptive group testing scheme with a low-complexity decoder in the N-asymptotic regime and present a binary compression scheme based on both a progressive encoding strategy and the combination of linear and OR-based encoding stages. Then, we present a novel adaptive compression scheme for group testing (chapter 5) and formulate it as a decision problem (chapter 6); by doing so we set up a framework to systematize the construction and

design of adaptive schemes for group testing. Furthermore we also propose tools to predict the performance these adaptive schemes in the N-asymptotic regime.

Finally, in part III we enclose the results of the research efforts we devoted to noiseless analog compression, which encompasses the noiseless compressed sensing (CS) problem [28, 43]. In chapter 7 we deduce a necessary and sufficient matrix condition to ensure that a noiseless CS scheme is lossless within the AC framework. Chapter 8 is devoted to present two novel decoding algorithms for noiseless compressed sensing that exploit that the AC signal model.

1.1 Signal model and compression scheme for analog compression

Figure 1.1 depicts the typical setup of an analog compression scheme:

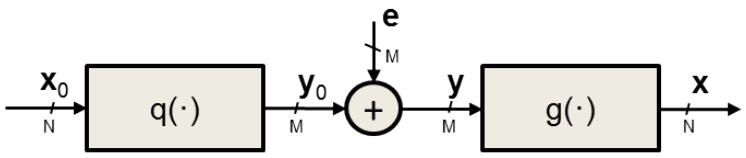


Figure 1.1: Analog Compression scheme

The original data vector $\mathbf{x}_0$ is composed of N independent realizations of a random variable X that is distributed according to a mixture of continuous and discrete distributions

$$\mathrm{f}_m(X) = \gamma\, \mathrm{f}(X) + (1-\gamma)\, \mathrm{P}(X) \tag{1.1}$$

where $\mathrm{f}(X)$ is a strictly continuous distribution, $\mathrm{P}(X)$ is a collection of mass concentrations and γ is the sparsity ratio.

The encoding function $\mathrm{q}(\cdot) : \mathcal{X} \subseteq \mathbb{R}^{\mathrm{N}} \to \mathcal{Y} \subseteq \mathbb{R}^{\mathrm{M}}$ maps a N-dimensional sequence $\mathbf{x}_0 \in \mathcal{X}$ to its low M-dimensional compressed representation $\mathbf{y}_0 \in \mathcal{Y}$, $\mathrm{M} < \mathrm{N}$.

Afterwards, the compressed sequence is transmitted through a channel to the decoder $\mathrm{g}(\cdot)$. In some situations the transmission of the compressed sequence is corrupted by noise ($\mathbf{e}$) so the decoder receives a degraded version of the compressed sequence $\mathbf{y}_0$, $\mathbf{y}$. Then the decoder $\mathrm{g}(\cdot) : \mathcal{Y} \subseteq \mathbb{R}^{\mathrm{M}} \to \mathcal{X} \subseteq \mathbb{R}^{\mathrm{N}}$ attempts to generate an estimate of the original source sequence $\mathbf{x}$.

1.2 Some compression problems

The scheme depicted in figure 1.1 setup encloses all the compression problems where we focused our research efforts. Next we review these problems and show that they are structurally related.

1.2.1 Noiseless compressed sensing

The scheme in figure 1.1 models the compressed sensing (CS) problem [28, 43, 44] whenever $\mathrm{P}(X)$ is a mass concentration at zero with weight $\mathrm{P}(X=0)=1-\gamma$ and $\mathrm{f}(X)$ is an arbitrary probability density function that characterizes the non-null components of vector of $\mathbf{x}_0$.

The CS problem arises in multiple applications where the selection of an appropriate basis leads to a sparse representation of the data. The CS problem establishes that this sparsity can also be exploited to recover a K-sparse signal $\mathbf{x}_0 \in \mathbb{R}^{\mathrm{N}}$ from a linear projection into a lower-dimensional space $\mathbf{y}_0 \in \mathbb{R}^{\mathrm{M}}$ under certain conditions from the l_0 minimization problem

$$\mathbf{x} = \underset{\mathbf{y}_0=\mathbf{A}\mathbf{x}_0}{\operatorname{argmin}} \ \|\mathbf{x}\|_0 \tag{1.2}$$

where $\mathbf{A}$ is a projection matrix, known as measurement matrix, and $\|\cdot\|_0$ is the l_0 pseudo-norm. The set $\mathcal{S}$ containing the positions of these non-zero components is known as the *support set* or *the sparsity pattern*, defined as $\mathcal{S} \triangleq \{i \in 1,...,\mathrm{N} : \ x_i \neq 0\}$, with $|\mathcal{S}| = \mathrm{K}$.

Note that there are some basic differences between the AC problem and the CS problem. In the classical CS problem the performance is measured in a worst-case scenario for all sparse $\mathbf{x}_0$ whereas in the proposed AC setup the $\mathbf{x}_0$ is randomly distributed and performance is measured probabilistically. Furthermore, in the classical CS problem the encoder is constrained to be a linear mapping characterized by a $\mathrm{M} \times \mathrm{N}$ matrix $\mathbf{A}$ known as measurement matrix whereas in the AC setup encloses also non-linear encoders. Finally, in the classical CS problem the decoder is assumed to be robust with respect to the noise observation, whereas the robustness conditions for the optimal decoders in AC setup are Lipschitz continuity for noiseless schemes and the minimum mean square error (MMSE) estimator in presence of noise (see [162] for further details).

In the state of the art several authors working within the analog compression framework refer to it as compressed sensing, since they apply the AC theory to solve the CS problem. Throughout the thesis we also refer to the AC framework as CS.

1.2.2 Binary source compression

The binary source compression problem can also be represented by the set up described in section 1.1, since it reduces to a classical binary source compression setup [139] in case that $\gamma = 0$ and $\mathrm{P}(X)$ has mass concentrations $\mathrm{P}(X=1)$ and $\mathrm{P}(X=0)$. In this case the AC problem reduces to the design of efficient schemes composed by an encoder and a decoder to represent a binary data source $\mathbf{x}_0 \in \{0,1\}^{\mathrm{N}}$ by means of a shorter sequence $\mathbf{y}_0 \in \{0,1\}^{\mathrm{M}}$, $\mathrm{M} < \mathrm{N}$.

Source compression schemes can be classified in two groups: schemes where the encoded sequence length M is fixed (fixed length codes) and those where M and the compression rate may depend on the source realization (variable length codes).

In this thesis we focus our attention on compression schemes where each component of vector $\mathbf{y}_0$ is generated by operating with some components of vector $\mathbf{x}_0$ indicated by the non-null components of a row vector of a sparse and binary matrix $\mathbf{A} \in \{0,1\}^{\mathrm{M}\times\mathrm{N}}$, so $\mathbf{y}_0 = \mathrm{q}(\mathbf{A},\mathbf{x}_0)$ where $\mathrm{q}(\cdot)$ models the operation.

In case that the encoding function generates the components of $\mathbf{y}_0$ with the logical XOR operation $\mathrm{q}_x(\cdot)$ the resulting scheme corresponds to a low density parity-check (LDPC) [129] code for source compression

$$\mathbf{y}_0 = \mathrm{q}_x\big(\mathbf{A},\mathbf{x}_0\big). \tag{1.3}$$

We have a low density product-check (LDPrC) code [106] for source compression when the encoding function $\mathrm{q}_a(\cdot)$ is the AND logical operator

$$\mathbf{y}_0 = \mathrm{q}_a\big(\mathbf{A},\mathbf{x}_0\big). \tag{1.4}$$

The OR-based encoder function $\mathrm{q}_o(\cdot)$

$$\mathbf{y}_0 = \mathrm{q}_o\big(\mathbf{A},\mathbf{x}_0\big) \tag{1.5}$$

is strongly related to the LDPrC codes since the negation of all the inputs and the outputs of an OR-gate results in an AND logical gate. The OR-based encoder is especially relevant for us because it models how the compressed sequence is generated in one of the main topics of the thesis: group testing (GT).

1.2.3 Group testing

In this problem each component of vector $\mathbf{y}_0$ represents to a test result obtained from a mixture of samples (e.g. blood, water, soil, etc) and aimed at detecting the presence of some undesired feature (e.g. a viral tests for blood testing) in the mixture of samples. The tests employed in GT are such that if the result of the test is non-defective all contributors can be labeled as non-defective and otherwise we know that one or more of the contributors are defective (e.g. infected blood).

The GT problem can be posed as a binary source compression problem where the population $\mathcal{P}$ is modelled by N independent realizations of a Bernoulli source P enclosed in vector $\mathbf{x}_0 \in \{0,1\}^{\mathrm{N}}$.

Each component of the realization represents a member of the population, the symbols 1 or 0 of each component indicate whether the subjects are either defective or non-defective and the operation of mixing the contributions of several subjects is modeled by the OR operator. Hence the compressed sequences $\mathbf{y}_0$ are generated as indicated by equation 1.5.

1.2.4 Sparse pattern recovery

This problem is strongly related to the CS sensing problem [59, 108, 110], since it is aimed to identify which subjects belong to the support set $\mathcal{S}$ of a K-sparse signal vector $\mathbf{x}_0 \in \mathbb{R}^{\mathrm{N}}$ with algorithms that require a small amount of computational resources (energy, memory, number of operations, etc).

The sparse pattern recovery problem naturally reduces to the group testing problem under the AC signal model in the noiseless setup when the measurement matrix is binary and sparse $\mathbf{A} \in \{0,1\}^{\mathrm{M}\times\mathrm{N}}$, since the signal vector $\mathbf{x}_0 \in \mathbb{R}^{\mathrm{N}}$ is composed by N realizations of a mixture data source with $\gamma > 0$. Under this restriction, the sparsity pattern recovery problem can be regarded as the reconstruction of a binary sequence $\mathbf{x}' \in \{0,1\}^{\mathrm{N}}$ of the same length than the sparse signal and with ones at the non-zero positions. Let us then define the following nonlinear operator:

$$\mathrm{q}(x) = \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{1.6}$$

which can be regarded as a one-bit quantization or threshold detection. Applying it to the measurements (element-wise), we observe that also the signal information in the linear projection is reduced to a binary basis, which corresponds to the sparsity pattern sequence $\mathbf{x}' = \mathrm{q}(\mathbf{x}_0)$:

$$\begin{aligned} \mathbf{y}' &\triangleq \mathrm{q}(\mathbf{A}\mathbf{x}_0) \\ \mathbf{y}' &= \mathrm{q}(\mathbf{A}\,\mathrm{q}(\mathbf{x}_0)) = \mathrm{q}(\mathbf{A}\mathbf{x}') \end{aligned} \tag{1.7}$$

This result holds because all the measurement of vector $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$ generated by the contribution of non-null components of vector $\mathbf{x}_0$ are also non-zero with probability 1, because the probability of drawing a number of independent non-zero samples from the source X, i.e. distributed according to a continuous mass distribution, that sums up to zero happens with probability 0. We can extend this results to sparse matrices $\mathbf{A}$ where the non-null components are i.i.d according to a continuous distribution without mass concentrations since in this case the values of the coefficients are all different. In the literature this problem is referred to as 1-bit compressed sensing [16] when the matrix is Gaussian or Radamacher.

The measurement operation in this equation (1.7) can be also expressed as the OR binary operation among sets of sparsity pattern bits. This result establishes a direct link between this

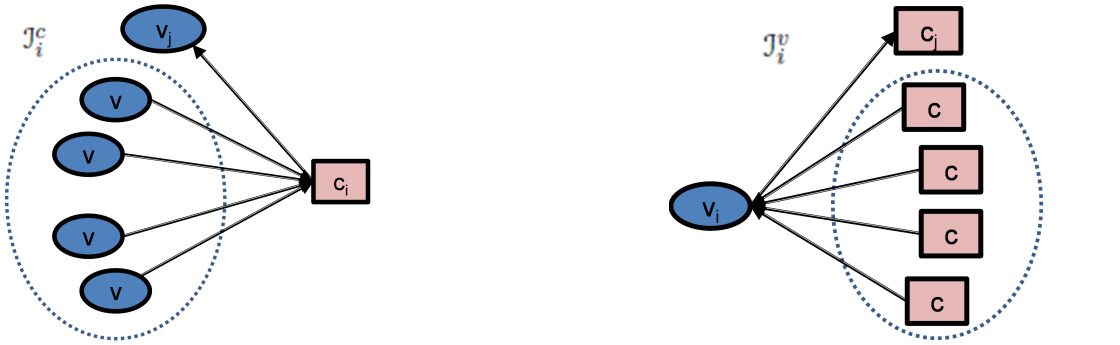
problem,the group testing problem and the OR-based encoder in equation (1.5).

1.3 Graph-based compression schemes

The compression schemes that employ the typical matrix-vector product or the encoding functions in equations (1.3), (1.4) and (1.5) have in common that can be described by a matrix. They differ only in the operations that the encoders employ to reconstruct the sequence $\mathbf{x}_0$ from its compressed version $\mathbf{y}_0$.

1.3.1 Graph representation

Matrices can be represented graphically as bipartite graphs (see [129]). Bipartite graphs depict the relations introduced by the matrix $\mathbf{A}$ between the components of vector $\mathbf{x}_0$ and vector $\mathbf{y}_0 = \mathrm{q}(\mathbf{A},\mathbf{x}_0)$. A bipartite graph $\mathcal{G}$ has two types of nodes: N variable nodes (one per entry of vector $\mathbf{x}_0$) and M check nodes (one per entry of vector $\mathbf{y}_0$). The connections (edges) between variable and check nodes are indicated by the non-null entries of the measurement matrix. A graph representing a real or complex valued matrix is also characterized by the value of the edges, whereas in binary graphs these values are not required since all edges are equal to 1.



(a) Neighborhood of a check node **(b)** Neighborhood of a variable node

Figure 1.2: Neighborhoods

Bipartite graphs belong to ensembles Γ (R,L,N) that are characterized by the number of variable nodes N and the check and variable node degree profile polynomials, R(d) and L(d) respectively, which define the fraction of check and variable nodes that have d edges in the graph. The performance of a randomly selected graph of an ensemble $\Gamma\big(\mathrm{R,L,N}\big)$ can be characterized by the average performance of all the graph in the ensemble in the N-asymptotic regime[1].

1 See for further details theorem 3.30 in [129].

In the literature the variable nodes connected to an arbitrary check node i are usually referred to as the neighborhood of the check node i, $\mathcal{I}_i^c$. Similarly the check nodes connected to a variable node i are referred to as the neighborhood of the variable node i, $\mathcal{I}_i^v$. Figures 1.2(b) and 1.2(a) depict the neighborhood of a variable and a check node, respectively. In these figures ellipsoids denote variable nodes whereas rectangles depicts check nodes.

In binary source compression setups graphs are in general sparse, i.e. the fraction d_v/N vanishes as N goes to infinite, whereas in compressed sensing setups they can be also dense.

1.3.2 Decoding algorithms

The graph representation of the problem allows to implement the decoder using iterative message passing algorithms where the values of the components represented by the variable nodes are retrieved employing message passing strategies. In these algorithms variable (check) nodes propagate messages to check (variable) nodes in their neighborhoods so each node computes the output messages only employing the information received from the nodes in its neighborhood.

In message passing strategies, check and variable nodes are activated alternatively. During each activation stage, each i-th variable (check) node updates the messages forwarded to each check (variable) node in its neighborhood employing only extrinsic information, e.g the i-th variable node v_i in figure 1.2(b) generates the message forwarded to the j-th check node employing only the messages received from all the other check nodes in its neighborhood $\mathcal{I}_i^v \setminus j$.

If the graph is a tree, in the binary case the decoder exploits the relations between the components of vector $\mathbf{x}_0$ and $\mathbf{y}_0$ described as a factor graph and employs the belief propagation (BP) algorithm [164], i.e. a message passing strategy over the graph representation of the matrix. The resulting problem is an approximation to the bit-wise maximum a posteriori (MAP) decoder when matrix $\mathbf{A}$ is sparse. An example of decoding belief propagation algorithm for channel decoding is found in [130].

In CS the MAP estimator can be implemented by the generalized BP decoder [164] that also poses the reconstruction problem as message passing strategy over the graph representation of the measurement matrix. Regretfully, under this approach the messages exchanged between check and variable nodes are probability density functions and, hence, the complexity becomes intractable. Several algorithms appeared in the literature aimed at approaching BP performance with a reduced complexity [13, 120].

As usual, there is a trade off between performance and complexity, since more accurate approximations require much more accurate models and hence more computations to track the evolution of the probability density functions.

The computational complexity depends of the amount of edges of the graph so when denser the graph more messages are propagated, so more computational power is required to update them. Thus, the computational complexity of the reconstruction algorithms depends critically on the sparsity (amount of edges) of the graphs. Sparse graphs usually perform O(N) edge upgrades whereas dense graphs update O(NM) edges.

It must be remarked that in the N-asymptotic regime if the BP-based message passing algorithms over sparse graph (with cycles) generates a solution with infinite reliability, it usually converges to the MAP solution of the problem because in the N-asymptotic case the length of cycles goes to infinite and, so the graphs converges to a tree.

CHAPTER 2

Binary source compression and other related problems

The binary source compression problem consists on the design of efficient schemes to represent a binary data sources $\mathbf{x}_0 \in \{0,1\}^{\mathrm{N}}$ by means of a shorter sequence $\mathbf{y}_0 \in \{0,1\}^{\mathrm{M}}$, $\mathrm{M} < \mathrm{N}$.

Shannon introduced the concept of information in [139] and established that sequences $\mathbf{x}_0$ composed by N independent realizations of a binary source X can be compressed in the N-asymptotic regime into a binary sequence $\mathbf{y}_0$ of fixed or variable length with an average compression rate r and reconstructed without losses whenever $r \geq \mathrm{H}(X)$, where $\mathrm{H}(X) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the entropy rate of the Bernoulli source. The source entropy divides the compression rate into two regions: one where lossless reconstruction is achievable, $r \geq \mathrm{H}(X)$ and another one where lossless reconstruction cannot be achieved, i.e. $r < \mathrm{H}(X)$.

We say that compression schemes with $r < \mathrm{H}(X)$ work in the *rate distortion* regime [138, 139], i.e. in this region the original sequences can be reconstructed with a distortion. The rate distortion zone is defined by the boundary[38]

$$r(D) \geq \mathrm{H}(X) - \mathrm{H}(D) \tag{2.1}$$

where D is the Hamming distance between the original sequence $\mathbf{x}_0$ and its reconstructed version $\mathbf{x}$.

In some situations it is possible to construct compression schemes with compression rates arbitrarily close to the entropy of the source [159], whereas in some other situations the lossless compression region of an specific compression scheme is structurally bounded away from the entropy bound due to physic constraints, e.g. delay constraints or encoder limitations. So in several compression problems the compression limits are unknown, e.g. in adaptive group testing schemes [10, 35, 77], and hence the source entropy rate is only a non-achievable upper bound.

We talk of *almost lossless compression schemes* whenever the encoder/decoder produces a rate of errors at the output of the decoder that vanishes with N. Furthermore, in some other situations it is not even possible to design almost lossless compression schemes, e.g. non-adaptive group testing [5].

This chapter is divided in three blocks. In section 2.1 we do a brief review of the literature related to different binary graph-based compression schemes we deal with. We devote section 2.2 to do an introduction to graph-based source compression schemes where both the sequence to be compressed and the codeword have fixed length. Finally section 2.3 is dedicated to introduce the reader to the main topic of part II: the group testing problem, which can be regarded as a graph-based source compression scheme with a non-linear encoder.

2.1 Binary graph-based source compression schemes

The objective of this section is to give a general overview of the literature related to binary source compression, stressing its relationship with our work.

In this section we briefly review the state of the art in source coding (2.1.1), in the sparse pattern recovery problem (2.1.2), and in the group testing problem (2.1.3).

2.1.1 Binary almost lossless source compression

In the literature there exist several variable rate source coding methods that can compress Bernoulli data sources up to its binary entropy: the Huffman codes[75], the entropy coding algorithm [56, 57, 133], the Lempel-Ziv based algorithms [170, 171].

Recently, a different approach appeared in the literature based on low density parity-check (LDPC) linear codes for fixed length coding. LDPC codes for channel coding were first proposed by Gallager [142] and later rediscovered by Mackay [100]. It is known that these schemes can asymptotically achieve the channel capacity in different conditions with an affordable complexity [142]. The key point of the LDPC-based codes is that its performance can be predicted using density evolution analysis [129], which attracted several research efforts that produced near-optimum codes. Afterwards, the duality between channel and source coding was successfully exploited to employ LDPC codes in source coding problems [20] and is referred to as syndrome coding [6].

Basically, the mathematical model this source compression scheme coincides with that one of transmitting the all zero codeword through a binary symmetric channel (BSC) channel with error probability p, i.e. the patterns of errors introduced by the BSC with probability p are equivalent to sequences generated by a Bernoulli source of probability p. Hence, the performance

of the source compression schemes depends critically on the capability of the matrix[1] to assign a different syndrome to each error pattern induced by the channel or, conversely, a different compressed sequence to each sequence generated by a Bernoulli source with entropy rate $\mathrm{H}(X)$.

In syndrome coding the number of sequences that are typically generated by the source, i.e. the *typical* set of source sequence, is approximately $2^{\mathrm{N}\,\mathrm{H}(X)}$. In case that $\mathrm{M} \geq \mathrm{N}\,\mathrm{H}(X)$ there is an overwhelming probability that matrix $\mathbf{A}$ behaves as an isomorphism for almost all the sequences in the typical set when matrix $\mathbf{A}$ is properly designed, since the encoder function performs a one-to-one mapping between the typical sequences and the code codewords [6, 20, 21]. The performance of these schemes is nearly optimal when the entropy rate of the source $\mathrm{H}(X)$ is high, but rate losses increase as the entropy of the source sequence decreases [19, 20]. This almost loss-less scheme can also be employed to deal with the problem of source compression with side information [95].

Fountain codes [99] are a different instance of linear codes for source coding. The special characteristic of this family of codes is that they are rateless, i.e. the length of sequence $\mathbf{y}_0$ is variable. Examples of these codes that have been applied to source coding are based on turbo-codes [71] or on the LT-codes [18, 96]. It has been shown in the literature that the fountain codes have empirically a good performance when dealing with the source coding problem [18, 96].

LT codes [96] is a special instance of fountain codes where the number of components of vector $\mathbf{x}_0$ involved in the computation of each component of sequence $\mathbf{y}_0$ d_c is chosen randomly from a check node degree profile distribution $\mathrm{R}(d)$. Afterwards, thosed_c components of vector $\mathbf{x}_0$ are chosen independently at random from a uniform distribution[2]. The decoding process starts when the number of compressed bits at the decoder side is sufficiently large. The design of distributions for the check node degree profiles is an open problem [33, 147].

Non-linear codes are potentially more powerful than linear ones [68, 69] since they include the latter as a particular case. This potential is of special interest in the case of non-uniform sources. The reason is that linear codes possess identical distance profiles for all codewords, while non-linear codes have different distance properties for different codewords. This fact can be exploited to guarantee better distance profiles for the most likely information sequences, which should lead to better performance.

In spite of this potential advantage, there has been relatively little work on non-linear codes, probably due to the fact that linear codes are known to be asymptotically optimum in channel

1 Referred to as parity check matrix in channel coding, matrix $\mathbf{A}$ here.
2 This process can be repeated at the decoder side to generate the parity matrix whenever the seed of the random number generator is known by both the encoder and the decoder.

coding for infinite block lengths. In this thesis, we focus on a special instance of non-linear codes known as low density product-check (LDPrC) codes [105, 106], since they use check nodes that perform the AND operation over their binary inputs. These codes are remarkable because their performance can be predicted by means of density evolution tool that helps to design them. Its extension to the OR-based check nodes is straightforward and, hence, it can be extended to Group testing schemes.

2.1.2 Sparse pattern recovery

As we showed in section 1.2.1, this problem is strongly related to the CS sensing problem [59, 108, 110, 123] because it is aimed at identifying the subjects that belong to the support set $\mathcal{S}$ of a K-sparse signal vector $\mathbf{x}_0 \in \mathbb{R}^{\mathrm{N}}$ with algorithms that require a low amount of computational resources.

The theoretical limits of the number of samples needed to recover the sparsity pattern perfectly or to achieve a certain error performance for a given (K,N) were given in [125]. In the noiseless case, the author states that *"methods and theoretical limits do not differ from CS"* which is intrinsically also a sparsity support recovery problem, i.e. that perfect reconstruction can be achieved with M approximately $\mathrm{O}\big(\mathrm{K}\log(\mathrm{N}/\mathrm{M})\big)$. Sparse pattern reconstruction algorithms are based on l_0 and l_1 minimization problems reviewed in chapter 3. The author in [124] explores the theoretical limits of sparse pattern recovery in the rate-distortion region.

2.1.3 Group testing

The group testing (GT) was proposed during the Second World War [51] and deals with the problem of sorting a subset of defective subjects $\mathcal{P}_D$ from the non-defective subjects $\mathcal{P}_N$ of a population $\mathcal{P}$ when the number of defective subjects K among the N subjects of the populations is low, $\mathrm{K} = |\mathcal{P}_D| < \mathrm{N}$. The relation between N and K can be either proportional or exponential with N, i.e. $\mathrm{K} = \beta\mathrm{N}$ or $\mathrm{K} = \mathrm{N}^{1-\beta}$ for $0 < \beta < 1$, respectively. In the exponential case the fraction of defective subjects in the population vanishes as N increases, i.e. $\mathrm{K} = \mathrm{o}(\mathrm{N})$, whereas in the proportional case $\mathrm{K} = \mathrm{O}(\mathrm{N})$. Research efforts in the literature are mainly focused on the exponential case whereas this thesis focus on $\mathrm{K} = \mathrm{O}(\mathrm{N})$.

GT exploits that when we select a subset of subjects of the population and perform a single test to all them together to detect a special characteristic or defect, if the result of the test is negative all the subjects contributing to the test belong to the set of non-defective subjects. Otherwise there is at least one defective subject in the sub-set of the population and more tests are required to identify the defective subjects. The noiseless GT setting focus on the cases where

the probability of false detection and miss detection are both equal to zero, i.e that the results of the tests are deterministic.

The group testing problem has been applied to a broad range of problems: industrial product testing [145], sequential screening of experimental variables [94], searching in storage systems [78, 89], multiple access communication [67, 160], data gathering in sensor networks [70], screening of clone libraries [12, 17], blood screening [152, 165], image compression [74], data streams testing [37] and diagnosis for digital logic systems [87], DNA screening [140], spectrum sensing [148, 169]. It is closely related to both compressed sensing (CS) [16, 64] and coding theory [103, 104].

Within the compression framework, the group testing encoding stage refers to the generation of the tests and can be regarded as the OR-encoding operation shown in equation (1.5), where a matrix $\mathbf{A} \in \{0,1\}^{\mathrm{M}\times\mathrm{N}}$ is a sort of generator matrix known as measurement matrix, each component of vector $\mathbf{x}_0$ represents a member of the population $\mathcal{P}$ and each component of vector $\mathbf{y} \in \{0,1\}^{\mathrm{M}}$ represents to the results of a test $\mathrm{q}_o(\cdot)$, which can be regarded as an OR-logical operation applied to the members of the population $\mathbf{x}_0$ indicated by the correspondent row vector of matrix $\mathbf{A}$.

Recently GT problem has attracted many research efforts mainly focused on both bounding the theoretical limits of the exponential approach of different GT schemes [2–4, 8, 10, 31, 36, 85, 90, 134] and the derivation of algorithms for GT [5, 35, 36, 119]. It must be noted that the author in [156, 157] focused exclusively on analyzing the performance of non-adaptive group testing scheme that employ regular bipartite graphs (pooling graphs) when $\mathrm{K} = \alpha\mathrm{N}$. The later group testing scheme can be regarded as an LDPrC code for source coding [105, 106] that employs a BP-based message passing algorithm for graphs with AND/OR check node operations to infer the labels of the subjects of the population. The authors of [105, 106] also developed a density evolution analysis tool that can characterize the performance of the BP-based message passing algorithm over a realization of any ensemble of bipartite graphs with AND/OR check node operation in the N-asymptotic regime. We devote section 2.2.2 to review the LDPrC codes.

GT schemes can be divided between adaptive [35, 77, 87, 119] and non-adaptive [5, 31] group testing schemes. In the non-adaptive GT all tests are executed in a single round of tests. The round of tests is decomposed in two phases: an acquisition phase where tests are executed following some rules and a second phase or decoding phase where an attempt to label each subject according to the information gathered from all the tests is performed.

A feature of AND/OR gates and so of non-adaptive GT schemes when $\mathrm{K} = \alpha\mathrm{N}$ is that after the

decoding phase a subset of subjects of the population may remain unlabeled [61][1], $\mathcal{U} \subseteq \mathcal{P}$, with a certain probability that decreases by increasing the number of tests or decreasing the amount of non-defective of the population, i.e. reducing the scheme sampling efficiency.

Adaptive group testing arises as an efficient solution to this problem, see [3, 73, 79, 119]. The main difference with non-adaptive GT is that adaptive schemes alternate several stages of acquisition and decoding phases. This fact enables that the information gathered after processing i stages of tests can be employed to design the next stage of tests.

The introduction of adaptivity into a group testing scheme also introduces an extra degree of difficulty, since the entity managing the adaptive group testing scheme must define a policy to design the structure of the next stage of tests taking into account somehow all the information gathered up to the moment. The rules to decide the structure of the latter stages of the adaptive scheme are crucial for the overall performance of an adaptive group testing scheme [3, 79]. In the literature, to the best knowledge of the author, the design of policies for adaptive group testing remains still as an open problem.

There are basically two different families of algorithms for adaptive group testing, each one follows a different strategy to overcome the policy design problems. Two-stage based algorithms [40] perform the tests in two sequential stages, so the second stage exploits the knowledge gathered from the first test stage. Random walk algorithms [35, 77, 87] generate tests one by one and once the subjects are labeled focuses the following tests on the subjects that remain unlabeled. Here, in this thesis, we devote part II to introduce a third approach: the concatenation of as many stages as required to retrieve the labels of all the subjects where after each stage only the unlabeled subjects are further tested [115, 119].

The characterization of the theoretical compression limits of adaptive group testing schemes when $\mathrm{K} = \mathrm{O}(\mathrm{N})$ remains as an open problem, i.e. within this situation, it is unknown the minimum number of tests required to reconstruct almost any population of N subjects and with K defective members. For $\mathrm{K} = \mathrm{o}(\mathrm{N})$ this problem has been attracting the attention of several authors [3, 10, 79, 134], mainly focusing their research efforts on bounding the performance of the best known algorithm [77][2], which nowadays requires $\mathrm{K}\log(\mathrm{N}/\mathrm{K}) + 2\mathrm{K}$ tests in the worst case scenario to label all the subjects of the population. Recent theoretical results show that the smallest possible number of tests with vanishing error probability for both adaptive and

1 The author states that codes with AND gates are only optimal when the amount of zeros goes to zero or, alternatively in GT, when the fraction of defective subjects of the population goes to zero.

2 Note that the authors in [35] presented in 2013 is the best performing algorithm up to the date which outperformed to [77] which was presented in 1972.

non-adaptive group testing schemes behaves as $\big(\mathrm{K}\log_2(\mathrm{N}/\mathrm{K})\big)(1+\mathrm{o}(1))$ for $\mathrm{K}=\mathrm{N}^{\beta}$ and $\beta<1/3$ [134][1].

2.2 Decoders for graph-based binary source compression schemes

In this section we review several decoding algorithms specifically aimed to retrieve the original sequence from compressed sequences generated as in equation (1.3), (1.4) and (1.5). First in section 2.2.1 we review the decoder for the XOR-based encoders. Then in section 2.2.2 we review the decoding algorithm for AND-based decoders. Finally in section 2.2.3 we review a hybrid compression scheme [105, 106] where the compressed sequence is partially generated XOR-based and AND-based encoders.

2.2.1 Low density parity-check codes

The LDPC decoders are based on iterative message passing algorithm on the bipartite graph representation defined by equations (1.3). There exist several message passing algorithms or decoding algorithms to implement LDPC decoder with different performance and complexity, e.g. the min-sum algorithm, the sum-product algorithm[2], or Maxwell based-decoder [107].

The average performance of the sum-product message passing algorithm over an specific ensemble of graphs in cycle-free in the N-asymptotic regime can be assessed employing density evolution analysis ([129]). This tool can be employed to estimate the average performance of any randomly selected graph of a ensemble $\Gamma\big(\mathrm{R},\mathrm{L},\mathrm{N}\big)$, since its performance coincides with the average performance of the graphs of the ensemble in the N-asymptotic regime [129]. We enclose the message passing update rules of sum-product algorithm for LDPC codes in appendix 2.A.1

2.2.2 Low density product-check codes

Low density product-check (LDPrC) codes for source compression proposed in [106] can be seen as a non-linear generalization of LDPC codes since i) the proposed non-linear codes can be graphically represented by means of a factor graph, ii) they can be decoded using belief propagation, and iii) their performance can be predicted, and the codes analyzed, using density evolution. Thus they can be easily designed when long block lengths are considered. This

1 Note that some authors refers to the model as $\mathrm{K}=\mathrm{N}^{\beta}$ [134] and some others as $\mathrm{K}=\mathrm{N}^{1-\beta}$ [10].

2 We use the sum-product algorithm to derive some results in our thesis. Furthermore we also use the specific implementation of the sum-product algorithm for binary erasure channel (BEC) [129] that inherits its performance with a reduced complexity.

distinguishes the proposed scheme from the few non-linear codes that were proposed for lossless and lossy compression [68, 69], which are not easy to analyze and generalize.

In these codes a binary compressed sequence $\mathbf{y}_0 \in \{0,1\}^{\mathrm{M}}$ of length M is generated from a sequence of source bits $\mathbf{x}_0 \in \{0,1\}^{\mathrm{N}}$ of length N independently distributed according to a Bernoulli distribution $\mathrm{P}(X=1)$. The AND-based encoder function $\mathrm{q}_a(\cdot)$ generates the coded bits with the AND-operator $\mathbf{y}_0 = \mathrm{q}_a\big(\mathbf{A},\mathbf{x}_0\big)$[1], shown in equation 1.4. $\mathbf{A} \in \{0,1\}^{\mathrm{M}\times\mathrm{N}}$ is sparse binary code matrix with rate $r = \mathrm{M}/\mathrm{N}$.

These codes pinpoint on the idea that when $p > 0.5$, i.e. the coded bits are generated by the AND operation, a coded bit equal to 1 implies that all the source bits that contributed to generate the coded bit are equal to one with infinite reliability. This fact causes that LDPrC codes usually converge fast when the code matrix is properly built. The drawback of these codes is that the source bits equal to zero can only be retrieved from coded bits equal to zero when all the source bits that contributed to generate the coded bit are equal to 1 but one. Thus, those coded bits generated by the contribution of two or more source bits equal to zero are not useful to retrieve any source bit.

We enumerate the message passing update rules for BP decoding of LDPrC codes in appendix 2.A.2.

2.2.3 Hybrid LDPC-LDPrC codes

Hybrid LDPrC-LDPC codes are constructed as a parallel concatenation of two block codes: a fraction α of coded bits is generated by a nonlinear low density product-check code and the remainder fraction, $1-\alpha$, by a linear low density parity-check code.

The linear block is encoded as in a standard LDPC code. Defining a generator matrix $\mathbf{G}$ of size $(1-\alpha)\mathrm{M} \times \mathrm{N}$ and the source sequence $\mathbf{x}_0 \in \{0,1\}^{\mathrm{N}}$ of length N, the encoding process can be expressed as:

$$\mathbf{c}_0 = \mathrm{q}_x\big(\mathbf{G},\mathbf{x}_0\big) \tag{2.2}$$

with $\mathbf{c}_0 \in \{0,1\}^{(1-\alpha)\mathrm{M}}$ the linear sequence. For the LDPrC code, he encoding process can be described in a compact form by defining a $(\alpha\mathrm{M}) \times \mathrm{N}$ generator matrix $\mathbf{P}$ whose (i,j) entry is 1 if

1 An OR-based encoding function is employed to compress sequences with $\mathrm{P}(X=1) \leq 0.5$ and the AND operator when $\mathrm{P}(X=1) \geq 0.5$. The OR-based encoding function will be referred to as $\mathrm{q}_o\big(\cdot\,,\cdot\big)$.

the information bit b_i is employed in the computation of the coded bit p_j, and 0 otherwise. We thus represent the encoding process as

$$\mathbf{p}_0 = \mathrm{q}_a\left(\mathbf{P},\mathbf{x}_0\right) \tag{2.3}$$

with $\mathbf{y}_0 \in \{0,1\}^{\alpha \mathrm{M}}$.

The LDPrC-LDPC codeword is built as $[\mathbf{c}_0 \;\; \mathbf{p}_0]$. Therefore, matrices $\mathbf{G}$ and $\mathbf{P}$ fully characterize the hybrid LDPC-LDPrC code. These matrices are sparse and have random appearance. Hence, if the codeword is long enough and the matrix has been properly designed, there will be few cycles in the graph and belief propagation will provide a quite accurate approximation to bit-wise maximum a posteriori (MAP) decoding. The message passing equations for the variable XOR-based check nodes are the same as in LDPC codes, whereas the message passing update rules of the AND-based check nodes are the ones of the LDPrC codes.

Hybrid LDPC-LDPrC codes are characterized by the degree profiles of the bit nodes (both for the linear and non-linear parts), of the parity check and product check nodes. The analysis tools presented in [106] focused on the design of regular codes by means of density evolution.

2.3 Graph-based group testing schemes

In this section we introduce the approach to group testing based on modern graph-based codes [129, 142]. The similarities between the noiseless GT problem and coding theory were noticed by Malyutov as early as the 70's [104]. It is straightforward to show that there exist strong analogy between the low-density-product-check (LDPrC) codes presented in subsection 2.2.2 and GT, since in LDPrC codes the coded bits are generated by the AND-operation while in Group testing tests are generated by the OR-operation.

We can try to infer which subjects of a population are defective from a block of tests $\mathbf{y}_0$ obtained as shown in equation (1.5) adapting the LDPrC message passing algorithm to our specific problem. We simply note that the equivalence between AND and OR-based gates, i.e an OR gate is an AND gate where the inputs and the output results are negated. Due to this, for GT the equivalent message passing algorithm for LDPrC codes is obtained by swapping all zeros by ones and the ones by zeros in vector $\mathbf{x}_0$ and $\mathbf{y}_0$, which translated to swap all signs of equations (2.11) in section 2.2.2. Alternatively we can also employ the *erasure decoding* (ED), an algorithm that we proposed in [116].

In this section we review a non-adaptive group testing scheme. First in subsection 2.3.1 we introduce the scheme. After that in 2.3.2 we review a low complexity decoding algorithm for

noiseless group testing. Finally in section 2.3.3 we give some insight of how the expected outcome of the group testing scheme depends of the encoder structure.

2.3.1 Problem statement

In the group testing problem an agent performs tests on groups of subjects of a population $\mathcal{P}$ to identify (label) the defective and the non-defective subjects of the population, $\mathcal{P}_D$ and $\mathcal{P}_N$. Initially the label of all subjects is unknown.

Throughout the section we model the N members of the population $\mathcal{P}$ as a binary source of data that generates sequences $\mathbf{x}_0$ with N independent realizations of a random variable with a non-vanishing probability of '1' equal to the average fraction of non-defective subjects in the population $\mathrm{P}(X=1)=\mathrm{P}(\mathcal{P}_D)$, so if there are $\mathrm{K}=\mathrm{P}(\mathcal{P}_D)\mathrm{N}$[1] defective members in the population and $\mathrm{N}-\mathrm{K}$ non-defective members in the population and hence $\mathbf{x}_0$ has K 'ones'.

The agent does the classification in two steps:

1. In a test generation or encoding step, the agent generates M tests y_j, $1 \leq j \leq \mathrm{M}$, each one of them including a subset $\mathcal{I}_j$ of the population subjects. The outcomes of the tests can be written as

$$y_j = \mathrm{q}_o\left(\mathbf{a}_j, \mathbf{x}_0\right)$$

   where $\mathrm{q}_o(\cdot,\cdot)$ is the OR-based encoder in equation (1.5), y_j is the j-th component of vector $\mathbf{y}_0$, $\mathbf{x}_0$ is a binary vector that represents the subjects of the population, and $\mathbf{a}_j$ is the j-th row vector of a binary and sparse matrix $\mathbf{A}$ whose non-null components are indexed in $\mathcal{I}_j$.

2. The agent, in a decoding step, employs a decoding function $\mathrm{g}(\cdot)$ to attempt to label all the subjects, i.e. tries to infer the values of the components of vector $\mathbf{x}_0$ as

$$\mathbf{x} = \mathrm{g}(\mathbf{y}_0, \mathbf{A})$$

   where $\mathbf{x}$ is the reconstructed sequence.

1 We assume throughout all the thesis that the amount of defective subjects is O(N), i.e. that $\mathrm{K}=\mathrm{P}(\mathcal{P}_D)\mathrm{N}$ instead of $\mathrm{K}=\mathrm{N}^{\mathrm{P}(\mathcal{P}_D)}$ for $0 \leq \mathrm{P}(\mathcal{P}_D) < 0.5$.

2.3.2 Erasure decoder for group testing

Now let us focus on the message passing strategy of the erasure decoding algorithm that we proposed in [116][1]. In the noiseless case, the inference rules to retrieve the value of the variable nodes connected to a OR-based check node are:

Remark 2.1 (Non-defective subjects inference)**.** *Whenever the result of a test is non-defective (zero) then all the subjects that contributed to it can be labeled as non-defective.*

This statement holds because in absence of noise tests do not fail and hence the probabilities of miss-detection and false detection are equal to zero. In this situation it is also possible to infer defective subjects from the vector of tests $\mathbf{y}_0$, since it holds that

Remark 2.2 (Defective subjects inference)**.** *The decoder can label as defective a subject contributing to a defective test whenever all the other subjects that contributed to it are already labeled as non-defective.*

Based on this dual behavior a very low complexity message passing algorithm for codes with OR-based check nodes can be envisaged. These message passing rules only consider three possible values for the messages exchanged between the nodes: 1 (defective subjects is perfectly known), 0 (non-defective subject is perfectly known) and '?' (for a subjects that cannot be labeled).

Data: $\mathbf{A}$, $\mathbf{y}_0$
Result: $\mathbf{x}$
1 **begin**
2 **for** $i \in \{1,\ldots,\mathrm{M}\}$ **do**
3 **Find non-defective**: If y_i is non-defective ('0'), all variable nodes in the neighborhood of y_i are labeled as non-defective ('0').
4 **end**
5 **Pruning**: Remove all variable nodes labeled as non-defective.
6 **for** $i \in \{1,\ldots,\mathrm{M}\}$ **do**
7 **Find defective**: If the tests y_i is defective ('1') and only one variable node is connected to it, label the variable node as defective ('1').
8 **end**
9 **end**

Algorithm 2.1: Refinement algorithm

1 We presented this algorithm in 2010. A different algorithm named combinatorial optimal matching pursuit (COMP) that exploits only remark 2.1 was presented in 2011 [31] and recently the authors in [5] presented an algorithm named definite defective (DD) that can be regarded as an instance of the Erasure decoder that employs the a priori information to decide the the label of the subjects that the ED cannot label. Th SS algorithm can be regarded as a sub-optimal approach to the LDPrC codes.

The decoder relies in the graph representation of the testing matrix and on remarks 2.1 and 2.2 and performs three steps described in algorithm 2.1. First, all check nodes (e.g. the j-th check node) communicate to all variable nodes in their neighborhood either their labeled status if they can infer them (as defective or non defective) or their "un-labeled" status (if they cannot determine whether that variable node represents a defective or non-defective subject). Second, all variable nodes send to the check nodes connected to them their label (if any of the check nodes has told them) or their "un-labeled" status (if they have not been told their label by any check node). Third and last, the check nodes use the messages they receive from the variable nodes in their neighborhood to attempt to label those subjects in their neighborhood that remain unlabeled. It can be easily seen that no further improvement is obtained by extending the decoder to more iterations.

In the first step, all variable nodes connected to a OR-based check node whose test is non-defective are labeled as non-defective due to remark 2.1. Then, these variable nodes are removed of the graph, doing so the knowledge of the non-defective subjects is employed to infer which subjects are defective, employing remark 2.2, since the check nodes that remain with a unique edge can label that variable node as defective.

As it can be deduced from this algorithm, there is a fraction of the measurements that never helps to recover any variable nodes: those defective tests with more than one edge after the pruning step.

Notice that this decoder does not require iterations nor arithmetic operations. The performance of such a simple decoder is clearly suboptimal, since the source statistics are not taken into account[1] and soft values are not exchanged between nodes. However, it has the nice feature that no errors are made. Based on the similarities of this decoder with that one employed in LDPC codes for the binary erasure channel, we refer to this low complexity decoder in the sequel as the "erasure decoder".

2.3.3 Impact of check node degree on subjects labeling

Let us analyze the behaviour of an OR-based check node of degree d_c. With this in mind, let us observe the amount of subjects of $\mathbf{x}_0$ that can be inferred from the value of tests $\mathbf{y}_0$ by the erasure decoder. Three different situations arise in the erasure decoder:

1 This algorithm is a low complexity implementation of the LDPrC code decoding algorithm described in section 2.2.2.

- Type I check nodes: All subjects connected to the check node are equal to '0' (non-defective). Then their OR operation is also '0' and with this test we can label the d_c subjects in the neighborhood of the check node as non-defective.
- Type II: All subjects but one are equal to '0', so the test is '1'. In this case, when the $d_c - 1$ subjects that are equal to '0' are labeled, the remaining subject can be also labeled as a '1'. However, the OR-based check node is not useful to label any non-defective subject even if the defective subject has already been labeled.
- Type III: At least two subjects are equal to '1'. In this case the only information that can be extracted from this test is that some subjects are defective. Perfect knowledge of the label of any of the contributors does not convey any further information on the labels of the other ones.

If vector $\mathbf{x}_0$ is composed by N independent realizations of a Bernoulli source with $\mathrm{P}(X = 1)$ and given a degree d_c of the check node, the fraction of check nodes that correspond to each of these three Types is determined by equations

$$\mathrm{P}(\text{Type I}) = \mathrm{P}(\mathcal{P}_N)^{d_c} \tag{2.4}$$

$$\mathrm{P}(\text{Type II}) = d_c\big(1 - \mathrm{P}(\mathcal{P}_N)\big)\,\mathrm{P}(\mathcal{P}_N)^{d_c-1} \tag{2.5}$$

$$\mathrm{P}(\text{Type III}) = 1 - \mathrm{P}(\text{Type I}) - \mathrm{P}(\text{Type II}) \tag{2.6}$$

Figure 2.1 shows the probability that the check node belongs to each one of the three Types listed above for the case $\mathrm{P}(X = 1)$=0.013. Note that the fraction of nodes of Type I is a decreasing function of the degree d_c, whereas the fraction of nodes of Type II exhibits a maximum that can be shown to appear when $d_c = -\log^{-1}\big(\mathrm{P}(X = 1)\big)$.

Figure 2.1 indicates that in the design of measurement matrices for group testing there is a trade-off in the selection of the degree of the OR-based check node. The higher the check node degree the higher the compression rate of the code but also the higher the probability of generating Type III check nodes and the lower the probability of generating Type I check nodes.

Taking into account this trade-off, it is apparent that the performance of the GT schemes can be improved if variable nodes that are connected to Type I check nodes have the smallest variable node degree (so no graph edges are "wasted" trying to label them again). Furthermore variable nodes representing defective subjects should be connected to Type II check nodes also with a small degree so as they can be recovered but they have a small contribution to generate tests that are '0'.

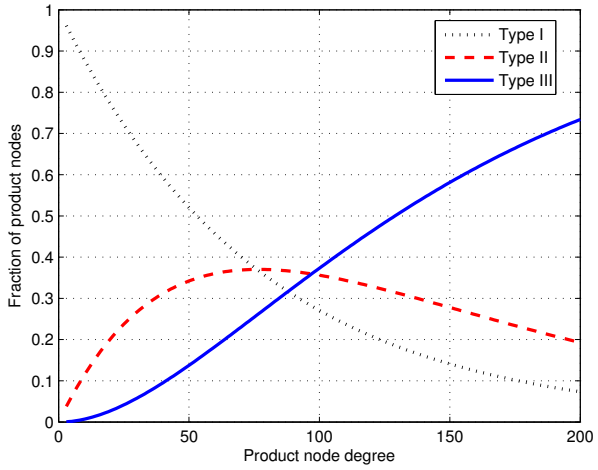


Figure 2.1: Distribution of the Type I, II and III check as a function of the node degree for $\mathrm{P}(X = 1)$=0.013

Appendix 2.A Belief propagation equations

2.A.1 Equations for XOR-based check nodes

There exist several message passing strategies to propagate the information under tree assumption in an iterative manner over the graph representing of a sparse matrix $\mathbf{A}$. Here we summarize the message passing update rules for bit-wise MAP decoding over binary-input memoryless channels defined by parity matrices $\mathbf{A}$ known as the *sum-product* algorithm. We also list the specific implementation of the sum-product for the binary erasure channel.

The basic structure of algorithms is: variable and check nodes are activated alternatively forwarding the extrinsic information to the nodes in their neighborhoods, i.e. the check/variable nodes forward a message to each variable/check node in their respective neighborhoods obtained from the fusion of all the messages received from the other nodes in their respective neighborhood. The information propagates throughout the graph as the number of iterations increases and hence the average bit-error probability is a non-increasing function with the number of iterations.

The messages exchanged by check and variable nodes are the log-likelihood ratios (LLR)

$$L = \log\left(\frac{\mathrm{P}(X=1)}{\mathrm{P}(X=0)}\right) \tag{2.7}$$

The update rules of the nodes are as follows: first each variable node 'v' propagates a different

message to each check node 'c$_j$' in their respective neighborhood $\mathcal{I}_v$:

$$L_{\mathrm{v}\to\mathrm{c}_j} = \sum_{\forall i\in\mathcal{I}_v\setminus j} L_{\mathrm{c}_i\to\mathrm{v}} + L_{app} \tag{2.8}$$

where $L_{\mathrm{v}\to\mathrm{c}_j}$ represents the message being updated, each $L_{\mathrm{c}_i\to\mathrm{v}}$ represent the incoming message received by the node 'v' from the check nodes 'c$_i$' and L_{app} is the a priori information available of the variable node 'v' (in case that the source is i.i.d depends only on the source asymmetry).

Similarly, in their turn each check node 'c' propagates to all variable nodes 'v$_j$' in in their respective neighborhoods $\mathcal{I}_c$, a message obtained as shown in equation (2.9)

$$L_{\mathrm{c}\to\mathrm{v_j}} = 2\tanh^{-1}\Big(\prod_{\forall i\in\mathcal{I}_c\setminus j} \tanh(L_{\mathrm{v}_j\to\mathrm{c}}/2)\Big) \tag{2.9}$$

Finally the estimation of value of the each bit is obtained as the sign of the message resulting from the addition of all the incoming messages

$$L_{\mathrm{v}_j} = \sum_{\forall i\in\mathcal{I}_{v_j}} L_{\mathrm{c}_i\to\mathrm{v}} + L_{app}. \tag{2.10}$$

The message passing algorithm for the BEC is a special instance of the sum-product algorithm. In this case the messages exchanged by the nodes are either $\pm\infty$ whenever the bit is known since the probabilities of equation (2.7) are ever either zero or one and the message L is zero whenever the bit is unknown or *erased*. The message update rules are:

$L_{\mathrm{v}\to\mathrm{c_j}}$ Propagate $\pm\infty$ if at least one of the messages $L_{\mathrm{c}_i\to\mathrm{v}}$, $i \in \mathcal{I}_v \setminus j$, is $\pm\infty$. Otherwise send a erasure message '?'.

$L_{\mathrm{c_j}\to\mathrm{v}}$ Propagate $\pm\infty$ if all the messages $L_{\mathrm{c}_i\to\mathrm{v}}$ from $\mathcal{I}_v \setminus j$ are $\pm\infty$. Choose the sign $+\infty$ if the number of $+\infty$ is even or $-\infty$ whenever the number of $+\infty$ is odd. Otherwise send the erasure symbol '?'.

2.A.2 Equations for AND/OR-based check nodes

As indicated in [105, 106], if we consider the case of a two-input AND operator $y = x_1 \cdot x_2$ and we denote by $L_{\mathrm{c}\to\mathrm{v}}$ and $L_{\mathrm{v}\to\mathrm{c}}$ the log-likelihood ratios (LLR) messages that go from the check node c to variable node v and vice versa then, we can write the decoding equations for this check

node of degree two as

$$L_{c\to \mathrm{v}_2} = \log\left(\frac{1+2\mathrm{e}^{L_{\mathrm{v}_1\to \mathrm{c}}+L_y}}{1+2\mathrm{e}^{L_{\mathrm{v}_1\to \mathrm{c}}}}\right) \tag{2.11}$$

For product nodes of higher degree, the messages can be computed recursively from the expressions above. The variable node message upgrade rules are the same than for message LDPC-based codes (see equation 2.8).

CHAPTER 3

Real data compression schemes

The goal of this chapter is to give to the reader a general overview of the noiseless compressed sensing problem [43]. This problem considers the estimation of an unknown sparse signal vector $\mathbf{x}_0 \in \mathbb{R}^{\mathrm{N}}$ from a vector of linear observations $\mathbf{y}_0 \in \mathbb{R}^{\mathrm{M}}$, $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$ where $\mathbf{A}$ is a $\mathrm{M} \times \mathrm{N}$ matrix known as measurement matrix. This is possible when $\mathrm{M} < \mathrm{N}$ because a small number K of elements of $\mathbf{x}_0$ are non-zero and matrix $\mathbf{A}$ is properly designed. In this case the estimate $\mathbf{x}$ with minimum l_0 norm that accomplishes the system of equations $\mathbf{y}_0 = \mathbf{A}\mathbf{x}$ coincides with the original signal vector $\mathbf{x}_0$.

While the minimization of the l_0-norm is a NP complex problem, the authors in [28, 44] advocated that this original optimization of the CS problem could be relaxed by using the l_1 norm. The key point to guarantee that the solution is the sparsest under this relaxation is the proper construction of the measurement matrix $\mathbf{A}$. The restricted isometry property (RIP) was found to be a sufficient design condition to guarantee the sparsity of the solution [11, 27]. Furthermore, the RIP condition also allowed any l_1-based reconstruction techniques to be robust to the presence of noise $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{w}$,

Recently the author in [161] posed the Compressed Sensing problem under the analog compression framework. The introduction of the analog compression framework for compressed sensing came along with new algorithms for compressed sensing that outperformed their predecessors. Since then several applications exploiting this new approach to the problem appeared in the literature: spectral sensing [155], parameter estimation [88, 136], sparse fast fourier transform (FFT) [72, 168], compressed image acquisition and compression [98, 146, 149].

The rest of the chapter is divided as follows: first in section 3.1 we enclose the theoretical limits of the CS problem within the analog compression framework. In section 3.2 we review the existing

decoding algorithms for CS whereas in section 3.3 we focus on different approaches to construct measurement matrices.

3.1 Information theoretic limits for compressed sensing

Here we summarize the compression limits of a noiseless analog compression setup when the encoder is linear. Further details and the extension to the noisy case can be found in [161–163].

The following definition gives the condition for a linear encoder to be considered well suited for noiseless compressed sensing [162]

Definition 3.1. *(Lipschitz continuity) A encoder function* $\mathrm{q}(\cdot)$ *is considered robust whenever it is L-Lipschitz continuous, i.e.* $+\infty > L \geq \mathrm{Lip}(\mathrm{q})$

$$\mathrm{Lip}(\mathrm{q}) = \sup_{\mathbf{x}\neq\mathbf{x}'} \frac{\|\mathrm{q}(\mathbf{x}) - \mathrm{q}(\mathbf{x}')\|}{\|\mathbf{x} - \mathbf{x}'\|}$$

where $\mathrm{Lip}(\cdot)$ *stands for the Lipschitz pseudo-norm of function* $\mathrm{q}(\cdot)$ *and* $\|\cdot\|$ *is the* l_2 *norm in this case.*

Note that in case of linear encoders $\mathrm{q}(\mathbf{x}_0) = \mathbf{A}\mathbf{x}_0$ the L coincides with the largest singular value of matrix $\mathbf{A}$.

Next we define the achievable rates of a compression scheme:

Definition 3.2. *(achievable rates of encoders) Let* $\mathbf{x}$ *be a random vector consisting of* N *independent realizations of* X*. Let* $\mathrm{q}(\cdot)$ *be Lipschitz continuous and* $\mathrm{g}(\cdot)$ *the* l_0 *decoder in equation 1.2. We refer to the minimum* ε*-achievable rate as the minimum of* $r > 0$ *such that there exists a linear encoder* $\mathrm{q} : \mathbb{R}^{\mathrm{N}} \to \mathbb{R}^{r\mathrm{N}}$ *and a decoder* $\mathrm{g} : \mathbb{R}^{r\mathrm{N}} \to \mathbb{R}^{\mathrm{N}}$*, such that*

$$\mathrm{P}\big(\mathbf{x} \neq \mathrm{g}(\mathrm{q}(\mathbf{x}))\big) \leq \varepsilon$$

for a sufficiently large N*. The minimum* ε*-achievable rate is denoted by* $r(X,\varepsilon)$*.*

The achievable rates are related to the information dimension of a random variable X, $\mathrm{d}(X)$, $0 \leq \mathrm{d}(X) \leq 1$ (see [126–128, 144] for further details). This measure of information characterizes the *amount of information* of a random variable X distributed according to a mixture distribution

$$\mathrm{d}(X) = \lim_{m\to\infty} \frac{\mathrm{H}(\lfloor mX \rfloor)}{\log(m)}$$

where $\lfloor \cdot \rfloor$ is the floor operator and $\mathrm{H}(\cdot)$ is the binary entropy of a probability mass function with m mass concentrations. If the limit does not exist the lim sup and the lim inf are called upper and lower information dimensions of X respectively, denoted by $\bar{\mathrm{d}}(X)$ and $\underline{\mathrm{d}}(X)$.

The author in [126] showed that the Renyi dimension of a random variable distributed according to a mixture of discrete and continuous distribution is $\mathrm{d}(X) = \gamma \mathrm{d}(X')$ where $0 < \mathrm{d}(X') \leq 1$ is the information dimension of a random variable distributed according to the continuous distribution. Examples of distribution with information dimension lower than 1 and equal to 1 are the Student distribution and the Gaussian distribution, respectively (see [128, 161] for further details).

The authors in [161] showed that the ε-achievable rate of a noiseless compression scheme coincides with the information dimension of the random variable

$$\mathrm{r}(X,\varepsilon) = \gamma$$

where $0 < \varepsilon \leq 1$ and X is a r.v. distributed according to a mixture of distributions with sparsity γ and the information dimension of the continuous part is $\mathrm{d}(X) = 1$. For $\varepsilon = 0$, i.e. zero distortion reconstruction, it holds that

$$\mathrm{r}(X,0) = \gamma + \mathrm{o}(1).$$

This result indicates that there exist lossless linear compression schemes in the noiseless setting, i.e. $\varepsilon = 0$ whenever for $r > \gamma$, i.e. at the output of the decoder appears $\mathbf{x}_0$ with probability 1. We say that a compression scheme of fixed rate r works in the almost lossless regime whenever ε vanishes with N. Otherwise, the scheme will be working on the rate distortion regime. We refer the reader to [161, 162] for further details. Note that this result leads to that the maximum compression efficiency δ is γ/r of a lossless compression scheme in the noiseless setting is $\delta = 1 - o(1)$.

This results apparently contradict the previously known results for lossless reconstruction in the noiseless setting in [43], where the authors showed that lossless reconstruction for linear encoders could only be achieved when the solution $\mathbf{x}$ of the l_0 minimization problem in equation (1.2) is unique and hence coincides with the $\mathbf{x}_0$. The authors showed that this result holds whenever the sequence sparsity (K) is less or equal to M/2 and the spark$(\cdot)$ of the measurement matrix, i.e. the minimum number of linear dependent columns of $\mathbf{A}$, is $\mathrm{spark}(\mathbf{A}) = \mathrm{M} + 1$[1]. We enclose this result in a lemma because we will refer to it later

1 See [43], theorem 3 and corollary 1.

Lemma 3.1. *Consider the noiseless compressed sensing setup presented in section 1.1 . Whenever the* $\text{spark}(\mathbf{A}) = \mathrm{M}+1$ *and* $\|\mathbf{x}_0\|_0 \leq \mathrm{M}/2$ *the* l_0*-based decoder in equation 1.2 has an unique solution.*

This result leads to a maximum sampling efficiency of $\delta = \mathrm{K}/\mathrm{M} = 1/2$ since the spark of matrix $\mathbf{A}$ is at most $\mathrm{M}+1$. So, apparently the condition in lemma 3.1 is only a sufficient condition for lossless compression. We focus chapter 7 on deriving a similar necessary and sufficient condition to guarantee that measurement matrix achieve a sampling efficiency close to 1.

3.2 Decoders for compressed sensing

Here we briefly review the most relevant reconstruction algorithms in the CS literature. We briefly introduce them and then we indicate both their computational complexity and performance.

3.2.1 l_1-based decoders

The authors in [28, 44] advocated that the original l_0 optimization of the CS problem in equation 1.2 could be relaxed by using the l_1 norm

$$\begin{aligned} \mathbf{x} &= \underset{\mathbf{x}}{\operatorname{argmin}} \quad \|\mathbf{x}\|_1 \\ \text{subject to: } & \mathbf{y} = \mathbf{A}\mathbf{x}. \end{aligned} \tag{3.1}$$

The signal reconstruction with l_1 algorithms has been proven to be robust to the presence of noise [91] under the RIP condition. The conditions to build RIP measurement matrices are given in section 3.3.1. The theoretical compression limits of l_1-based CS reconstruction methods, known as phase transitions, where fundamentally described in [49]. All these facts allowed the development of algorithms for robust compressed sensing and applied to problems that had been previously investigated by the statistical and machine learning communities. These approaches can be seen as a generalization of the problem given in equation (3.1) for noisy scenarios. Some of these approaches are the *Dantzig Selector* program that solves the linear optimization [29]

$$\begin{aligned} \mathbf{x} &= \underset{\mathbf{x}}{\operatorname{argmin}} \quad \|\mathbf{x}\|_1 \\ \text{subject to: } & \left\|\mathbf{A}^T(\mathbf{y}_0 - \mathbf{A}\mathbf{x})\right\|_\infty \leq b. \end{aligned} \tag{3.2}$$

for a given parameter b and the Basis Pursuit Denoising problem [34]

$$\begin{aligned} \mathbf{x} = \underset{\mathbf{x}}{\text{minimize}} \quad & \|\mathbf{x}\|_1 \\ \text{subject to:} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq b. \end{aligned} \tag{3.3}$$

3.2.2 Greedy algorithms

Greedy algorithms are a low complexity alternative to the l_1-minimization approach. These algorithms were first proposed to solve the optimization problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \\ \text{subject to:} \quad & \|\mathbf{x}\|_0 \leq \mathrm{K}. \end{aligned} \tag{3.4}$$

This problem is not convex, as the sub-space of $\mathbb{R}^{\mathrm{N}}$ delimited by $\|\mathbf{x}\|_0 \leq \mathrm{K}$ is not convex.

The first proposed greedy algorithm to solve this problem was the iterative hard thresholding (ITH) algorithm [15, 60], which can be seen as an special instance of a gradient projection algorithm [52, 58, 66]. It can run with $\mathrm{O}(\mathrm{N}\log(\mathrm{N}))$ computational complexity and requires a RIP matrix with $\mathrm{O}(\mathrm{K}\log(\mathrm{N}/\mathrm{K}))$ samples to find the original signal vector $\mathbf{x}_0$. In spite of being computationally affordable, the ITH algorithm performs worse than the l_1-minimization algorithms [101].

After the introduction of the ITH, more efficient greedy algorithms were proposed: the matching pursuit (MP) [102], the orthogonal matching pursuit (OMP) [65, 151], the iterative soft thresholding (IST) [55], the regular orthogonal matching pursuit (ROMP) [112], the Subspace Pursuit [39] or the compressed sampling matching pursuit (CoSaMP) [111]. The drawback of all these algorithms is that they require to know the signal sparsity K a priori. In order to solve this problem, the author in [42] proposed the sparsity adaptive matching pursuit (SAMP) and the authors in [101] proposed the tuned two-stage thresholding (tuned TST) that was reported to outperform the preceding algorithms. All them require $\mathrm{O}(\mathrm{K}\log(\mathrm{N}/\mathrm{K}))$ samples to reconstruct the original sequence when RIP matrices are employed.

3.2.3 Bayesian based decoders

The best performing algorithms for CS are based on an approximation to the maximum a posteriori (MAP) [1, 84, 109] based on belief propagation (BP) [13, 121]. Under this approach it is assumed that both the signal vector $\mathbf{x}_0$ and the noise vector $\mathbf{w}$ are sampled from some

probability distributions and the goal is to estimate $\hat{\mathbf{x}}$ though a maximum a posteriori optimization as shown next

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \, \mathrm{P}(\mathbf{x} \mid \mathbf{y}) = \underset{\mathbf{x}}{\operatorname{argmax}} \, \mathrm{P}(\mathbf{y} \mid \mathbf{x}) \, \mathrm{P}(\mathbf{x}) \tag{3.5}$$

where $\mathrm{P}(\mathbf{y} \mid \mathbf{x})$ models the sampling process and $\mathrm{P}(\mathbf{x})$ is the prior distribution. The performance of the schemes based on the MAP estimator depends critically on the measurement matrix, on the approximations made to reduce the computational complexity of the problem and on the selected prior.

The BP decoder poses the reconstruction problem as message passing strategy over the graph representation of the measurement matrix. Several algorithms appeared in the literature aimed at approaching BP performance with a reduced complexity, since in this case the computational complexity is unaffordable, as stated in section 1.3.2.

Next we review some algorithms aimed at solving the MAP problem with different approximations and priors. They are grouped by the prior they use.

Laplacian prior. The Lasso problem

The authors in [84, 150] showed that the Bayesian CS approach reduces to a least absolute shrinkage and selection operator (LASSO) problem when the noise is white Gaussian with zero mean and the a priori is assumed to be a Laplacian density function. This problem solves the l_1 regularized regression problem

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \, \|\mathbf{x}\|_1 \, . \tag{3.6}$$

for a proper selection of parameter λ. Thus the RIP matrices suffices to guarantee the convergence of the process in the noisy setting with $\mathrm{O}(\mathrm{N}^{3/2}\mathrm{M}^2)$ computational complexity[1].

In [13] it was empirically shown that the Compressed Sensing Belief Propagation algorithm and a properly generated sparse Radamacher matrices with $\mathrm{O}(\mathrm{N}\log(\mathrm{N}))$ measurements suffices to reconstruct almost any K-sparse signal with $\mathrm{O}(\mathrm{N}\log^2(\mathrm{N}))$ computational complexity. The author in [1] proposed the sum-product expectation maximization (SuPrEm) algorithm that it was empirically shown to exhibit good performance with $\mathrm{O}(\mathrm{N})$ computational complexity even though it does not have any guarantee to converge.

1 Recently a method to solve the LASSO by quadratic programming was proposed in [158].

3.2.4 Approximate message passing algorithms

The family of approximate message passing (AMP) algorithms can be divided in two sub-families: some authors [47] aimed their research efforts to solve the MAP problem with a Laplacian prior. The authors in [120, 122] presented the generalized approximate message passing (GAMP), a systematic method to implement the MAP estimation problem with arbitrary priors.

Note that all the AMP algorithms assume that the signal is a random realization from a mixture source and, hence, they are algorithms based on the AC framework.

Approximate message passing for LASSO

The first proposed low complexity algorithm to solve the Bayesian problem with Laplace prior was the AMP [47, 48]. As its name states, the authors posed this algorithm as a message passing strategy over a graph representation of a dense matrix where messages are parametrization of probability density functions.

It has been proven theoretically that this algorithm converges fast to the LASSO solution with $\mathrm{O}\big(\mathrm{K}\log(\mathrm{N}/\mathrm{K})\big)$ samples for RIP matrices, e.g. Gaussian distributed measurement matrix, whereas it requires only $\mathrm{O}\big(\mathrm{K}\big) + \mathrm{o}\big(\mathrm{N}\big)$ when the measurement matrix is band diagonal, as stated in section 3.3.3 (see [46]).

It is possible to predict the performance of a compressed sensing scheme composed by a band-diagonal matrix and an AMP decoder using the state evolution analysis presented in [46, 83].

The computational complexity of the AMP algorithm is O(MN), since the band-diagonal measurement matrix is also a dense matrix.

Generalized approximate message passing

The GAMP [88, 120] is aimed to solve the MAP approach irrespectively of the a priori distribution of the original signal. The authors posed the algorithm as a message passing strategy over the graphical representation of the matrix which can be either dense or sparse. As opposed to AMP algorithm, GAMP can also be applied to complex sources.

It has been empirically shown that this algorithm converges to the optimal solution in some cases but, to the best of our knowledge, the convergence of the algorithm has not been theoretically proven yet. Further details on the implementation of the algorithm can be found in[120, 122].

The computational complexity of the GAMP is also O(MN) when the measurement matrix is dense. Otherwise, when the matrix is sparse, its computational complexity is O(N).

Estimate-maximize approximate message passing with Gaussian-Bernoulli prior

The estimate-maximize Gaussian-Bernoulli approximate message passing (EM-GB-AMP) algorithm was develop in parallel and independently to the GAMP and AMP. It employs a message passing strategy over a graphical representation of the matrix (even in case of being non-sparse).

This algorithm assumes that the probability density functions exchanged by the nodes are Gaussian and, hence, describes them with two parameters (the mean and the variance) and it also assumes that the prior distribution is a Gaussian-Bernoulli mixture distribution. Furthermore it assumes that the sparsity γ and the variance of the non-null components of the vector $\mathbf{x}_0$ are unknown, so it learns them iteratively.

In chapter 8 we use this algorithm as a performance benchmark. We decided to do so because when this algorithm converges it manages to generate estimates $\mathbf{x}$ of the original sequence $\mathbf{x}_0$ so as its distortion $\|\mathbf{x}_0 - \mathbf{x}\|_2^2 \to 0$ as the number of iterations increases, whereas the AMP algorithm does not converge to zero distortion. Further details on the EM-GB-AMP algorithm can be found in [135] and [154].

In this case the computational complexity of the algorithm is also O(MN) when the measurement matrix is dense and O(N) when the matrix is sparse.

Note that empirical results show that the AMP-based algorithms can outperform the l_1-based algorithms [92, 135], since they outperform their theoretical limits. The reader should note that the different performance of either the GAMP and the GB-EM-AMP and the AMP for LASSO cab be attributed to both the choice of the Laplacian prior and the approximations performed.

3.2.5 Verification algorithms for noiseless compression

The verification algorithms (VA) are a family of local message passing strategies that can be regarded as a low-complexity approach to Bayesian compressed sensing because they structurally assume that the signal to be reconstructed has several zero components and, hence, that it is distributed according to a mixture distribution with a mass concentration at zero.

VAs approximate the density functions that nodes exchange under the generalized BP by probability mass functions (with one mass concentrated either at zero or at a non-zero value) to indicate that the mass concentration is verified, i.e. has infinity reliability, or not.

VAs exploit that the graph is sparse and that vector $\mathbf{x}_0$ is distributed according to a mixture source, as in equation (1.1). Due to the latter fact all the non-null coefficients of the signal vector $\mathbf{x}_0$ must be different and the probability that a sum of a given set of non-null coefficient is equal to another non-null coefficient of the signal vector is also equal to zero.

The main features of the verification algorithms are that (i) variable nodes can infer their values with infinite reliability in some situations, (ii) they can be implemented with O(N) computational complexity and (iii) their performance does not depend of the matrix coefficient values whenever the corresponding graph is free of length-four cycles [53, 166].

The best performing verification-based algorithm for CS is the node-based verification algorithm (NB-VA) [54], since it gathers all the features of this family of algorithms. In chapter 8 we employ this algorithm as a performance benchmark for our algorithms for noiseless CS. Furthermore, this algorithm is the departing point for enhanced verification algorithm (EVA) that we presented in [117] and we introduce in chapter 8. For completeness the NB-VA is described concisely in chapter 8.

The authors in [54] developed recently a density evolution analysis that predicts the performance of a compression scheme composed by a sparse matrix and the NB-VA decoder. The computational complexity of this algorithms is O(N), since they operate over sparse measurement matrices.

Empirical results show that this family of algorithms outperform the l_1-based algorithms in the noiseless setup in spite of their reduced complexity [117]. Furthermore, in section 8.3 we evidence that in some situations verification-based algorithms have performance similar to that one of AMP-based algorithms.

3.3 Encoders for compressed sensing

Here we review the state of the art in measurement matrices for the compressed sensing problem. We have grouped them into four different architectures: random (full) matrices with i.i.d. coefficients (section 3.3.1), random (possibly sparse) matrices that can be described as a graph (see section 3.3.2), structured (possibly sparse) matrices (section 3.3.3) and matrices whose structure stems from an algebraic coding scheme (section 3.3.4).

3.3.1 Restricted isometric property for l_1-based compressed sensing

With the proposal of l_1 relaxation of the compressed sensing problem several matrix properties were introduced that guarantee that the reconstruction algorithms lead to a unique solution [25, 26, 28, 44, 45].

The authors in [26] introduced the *Restricted Isometry Property* (RIP) condition that is strictly required to ensure that the l_1 reconstruction process does not fail. It states that any $\mathrm{M} \times \mathrm{N}$

matrix $\mathbf{A}$ is a (K,ε) – RIP matrix, whenever for every integer K and $0 \leq \varepsilon \leq 1$, any K-sparse vector $\mathbf{x}$ satisfies that

$$(1-\varepsilon)\left\|\mathbf{x}\right\|_2 \leq \left\|\mathbf{A}\mathbf{x}\right\|_2 \leq (1+\varepsilon)\left\|\mathbf{x}\right\|_2 .$$

In words, these property deals with the rank of the adjacent matrices $\mathbf{A}'$ that compose $\mathbf{A}$. The RIP condition guarantees a robust compressed sensing reconstruction for l_1-based algorithm [24, 44]. RIP matrices need $\mathrm{O}(\mathrm{K}\log \mathrm{N}/\mathrm{K})$ samples to guarantee the robust reconstruction of any K-sparse sequence.

In parallel, the authors in [11] showed that a $\mathrm{M}\times\mathrm{N}$ Gaussian sensing matrix (a matrix with its i.i.d. entries from the distribution $\mathcal{N}(0,1/\mathrm{M})$) or a Radamacher sensing matrix (a matrix whose entries are $\pm 1/\sqrt{\mathrm{M}}$ with probability 0.5) with $\mathrm{O}\left(\mathrm{K}\log \mathrm{N}/\mathrm{K}\right)$ samples can be employed jointly with the l_1-algorithms to estimate the original sequence in the presence of noise. It must be noted that these matrices guarantee that the support of the estimated sequence $\mathbf{x}$ coincides with the support of the original one $\mathbf{x}_0$.

Afterwards, several greedy algorithms (see section 3.2.2), and l_1-based algorithms (see section 3.2.1) were proposed to deal with both the noiseless and robust CS with differences in the required number of measurements required to reconstruct the original sequence and different computational complexity (see [7, 22] for a comparison of the performance of several different algorithms for different RIP matrices). Analytical bounds for the performance of l_1-based robust reconstruction algorithms with RIP matrices are given in [41, 49, 82, 114, 125].

3.3.2 Random graphs

We already introduced the basic graph notation in section 1.3. Here we review two different instances of graph-based encoders for compressed sensing: the sparse random graph-based encoders and the expander-graph based encoder.

Sparse graphs

The authors of [53, 54, 93] employed the density evolution analysis of LPDC codes for the binary erasure channel (BEC) in the N-asymptotic regime [97] to obtain the BP-threshold of ensembles of graphs for noiseless Compressed sensing γ^{BP}, i.e. the maximum sparsity so the linear encoder characterized by the ensemble of graphs behaves as an isomorphism. This result arises from the duality between a noiseless compressed sensing scheme where the support set is known and a

LDPC code correcting a codeword conveyed through a BEC (see [129] for further details on BP decoding for the BEC)[1].

Furthermore, the authors in [53] also showed that the performance in terms of γ^{BP} of any graph picked at random from an ensemble of graphs concentrates around to the ensemble average[2]. Hence, the performance of a ensemble of graphs can be characterized by its γ^{BP}.

The BP-threshold γ^{BP} of a ensemble of graphs $\Gamma\big(\mathrm{L},\mathrm{R}\big)$ provides insight into the reconstruction capabilities of the matrices of the ensemble because it indicates that the maximum achievable efficiency of a compression scheme that employs a matrix of the ensemble is $\delta^{BP}{=}\,\gamma^{BP}/r$ [53, 129].

Expander graphs

Expander graphs [143] have also been employed in compressed sensing. The graph $\mathcal{G}$ is a $(\mathrm{K},\varepsilon,d_v)$-expander whenever it has a regular variable nodes degree distribution[3], so each variable node is connected to d_v check nodes, and any sub-set $\mathcal{S}$ of variable nodes $\mathbf{x}$, $|\mathcal{S}| \leq \mathrm{K}$, with a neighborhood of the support set $\mathcal{S}$, $\mathcal{X}'$ has size $|\mathcal{X}'| > (1-\varepsilon)d_v|\mathcal{S}|$.

The authors in [14] showed that there exist $(\mathrm{K},\varepsilon,d_v)$-expander graphs with $d_v = \mathrm{O}\big(\log(\mathrm{N}/\mathrm{K})/\varepsilon\big)$ and $\mathrm{M} = \mathrm{O}\big(\mathrm{K}\log(\mathrm{N}/\mathrm{K})/\varepsilon\big)$ for $1 \leq \mathrm{K} \leq \mathrm{N}/2$. The author in [32] stated that binary graphs do not satisfy the RIP property stated in section 3.3.1. In spite of this, the authors in [14] derived a similar property in the l_1 norm (known as RIP-1) that guarantees the uniqueness of the sparse representation when a expander graph is employed. The RIP-1 property says that for any $(\mathrm{K},\varepsilon,d_v)$-expander graph $\mathcal{G}$ that represents matrix $\mathbf{A}$, it holds that:

$$(1-2\varepsilon)d_v \left\|\mathbf{x}\right\|_1 \leq \left\|\mathbf{A}\mathbf{x}\right\|_1 \leq d_v \left\|\mathbf{x}\right\|_1 \tag{3.7}$$

for any $\mathbf{x} \in \mathbb{R}^{\mathrm{N}}$. From this result it can be deduced that any K-sparse sequence can be reconstructed whenever the measurement matrix does not map any two K-sparse sequences to the same measurement vector and this happens whenever $\mathrm{M} = \mathrm{O}\big(\mathrm{K}\log(\mathrm{N}/\mathrm{K})\big)$. Some matrix constructions that satisfy this condition are proposed in [14], [81], [30].

Although sparse graphs require $\mathrm{M} = \mathrm{O}\big(\mathrm{K}\log(\mathrm{N}/\mathrm{K})\big)$, this number also shows a strong dependence on the degree of the check nodes d_c. The authors in [132] showed that d_c is $\mathrm{O}\big(\log(\mathrm{N}/\mathrm{K})\big)$ for

1 Note that, when the support set is known, the decoder knows the value of some components (those that are equal to zero) but it does not know the value of the non-null components. In a BEC setup, at the output of the channel, some bits of $\mathbf{y}$ have been erased and hence are unknown, whereas the value of the remaining bits is known.

2 Similar to theorem 3.30 in [129].

3 $\mathrm{L}(d_v)$ has only one non-null coefficient at d_v.

LDPC-based graphs (without any restriction in L(d)). Recently,[117], we gave a closed-form expression that approximates the R(d) that was empirically obtained in [132]. Although the measurement matrices obtained in [117] are not strictly expander graphs, as the distribution of the variable nodes is not regular, it was shown empirically that it is possible to construct binary matrices with $2K < M < 3K$ that ensure that a large fraction of K-sparse sequences can be recovered.

3.3.3 Structured (sparse) matrices

Recently the authors in [93] proposed a structured construction of the measurement matrix, referred to as coupled or band-diagonal matrices, to deal with the noiseless compressed sensing that empirically showed a good performance. As their name indicates, these matrices have non-null components concentrated around their diagonal. These coefficients are i.i.d. from Gaussian distributions with different variances (see [46, 83] for further details on how to construct them).

Later empirical results in [92] showed that band diagonal matrices in some cases allow to some decoders to achieve the lossless reconstruction in the noiseless setting with sampling efficiencies δ close to 1.

The authors in [46] studied rigorously the performance of a compression scheme composed by this kind of measurement matrices and the appropriate decoding algorithm and showed that their approach was successful as long as the compression rate r exceeded the (upper) Renyi information dimension of the signal $r > \bar{d}(x)$. In other words a K-sparse sequence $\mathbf{x}$ of length N whose distribution has Renyi information dimension $0 < \bar{d}(x) \leq 1$ can be reconstructed with high probability from $\bar{d}(x)\mathrm{N} + \mathrm{o(N)}$ samples, which for sparse structured sources with $\bar{d}(x) = 1$ leads to sample efficiencies $\delta = 1 - \mathrm{o}(1)$ for a sufficiently large N.

3.3.4 Reed-Muller compressed sensing

Reed-Muller compressed sensing considers the construction of measurement matrices where the matrix columns are obtained from exponentiating codewords in the quaternary Delsarte-Goethas (DG) code. Due to this the coherence of the columns inherits the properties of these algebraic codes. The author of [80] proved that CS with DG measurement matrices and the *one-step thresholding* (OST) algorithm [9] has almost the same performance as the recovery with LASSO and RIP matrices. It must be noted that this chirp reconstruction algorithm can also deal with the reconstruction problem with $\mathrm{O}(\mathrm{K}^2)$ computational complexity [7, 23] as long as the sparsity of the original signal is not very high.

PART II

Binary source compression and group testing

CHAPTER 4

Fixed-rate OR-based schemes for binary source compression

This chapter is devoted to characterize the performance of fixed length schemes for non-adaptive GT (in other words, a binary source compression with AND/OR-based encoders) and to present a first attempt to introduce adaptivity within the binary source compression framework. With this goal in mind, first in section 4.1 we review the signal model and introduce some notation that will help us to focus section 4.2 on analyzing deriving bounds on the average performance in the N-asymptotic limit of a generic non-adaptive graph-based group testing encoder $\mathrm{q}_o(\cdot)$ where the relation between the components of $\mathbf{x}_0$ and $\mathbf{y}_0$ are described by a random sparse matrix $\mathbf{A}$ that belongs to the ensemble Γ(R,L). The resultant analysis is similar in spirit to the density evolution for the binary-erasure channel [142].

Afterwards, in section 4.3 we focus on characterizing the compression limits of non-adaptive group testing scheme when the erasure decoder introduced in section 2.3.2 is employed. Regretfully, in this case the analysis is more complex because the operative of check nodes introduces an asymmetry in the decoding process, reflected in remarks 2.1 and 2.2 in section 2.3.2.

Finally, in section 4.4 we propose an adaptive scheme within the binary source compression framework that generates a fraction of the compressed sequence with AND-gates and the other fraction with XOR-gates. We presented this almost lossless compression scheme in [116] as a first approach to overcome the structural limitations of AND/OR encoder.

4.1 Previous concepts

4.1.1 Population model

In this chapter we consider the signal model introduced in chapter 1, i.e. that the signal vector generated from N independent realizations of a random variable P with $\gamma = 0$ and two

mass concentrations, {0,1}. Throughout all this section we employ the group testing setup, so $\mathrm{P}(P=1) < \mathrm{P}(P=0)$, we refer to the '1' and '0' as defective and non-defective and assume that K is approximately $\mathrm{N} \cdot \mathrm{P}(P=1)$, since we assume that N is large. We consider the OR-based sparse encoder described by function $\mathrm{g}_o(\cdot)$ in equation (1.5)[1] and the matrix $\mathbf{A}$ or alternatively its ensemble. Note that $'?'$ refers to unlabeled subjects.

4.1.2 Set description of the non-adaptive group testing problem

Here we describe all the sets required to track the evolution of the non-adaptive group testing scheme reviewed in section 2.3. All these sets contain finite many subjects of the population and help us to model the source realization $\mathbf{x}_0$ and its estimate $\mathbf{x}$. We employ the counting measure $|\cdot|$ to refer to the finite amount of subjects indexed in the sets.

First we formally define

Definition 4.1 (Labeled subject of a population)**.** *Consider a sparse matrix* $\mathbf{A}$*, an OR-based encoder, a sequence* $\mathbf{x}_0$ *and the graph representation of* $\mathbf{y}_0 = \mathrm{q}_o(\mathbf{A},\mathbf{x}_0)$*. We say that a subject (variable node) is labeled whenever its label can be inferred with infinite reliability by at least a check node in its neighborhood* $\mathcal{I}^v$*.*

So we only consider as labeled to those subjects of the population whose label can be inferred with zero error probability.

Vector $\mathbf{x}_0$ can be described by three sets: the sets that enclose the defective and non-defective subjects of the population and the population set as $\mathcal{P}_D \triangleq \{\forall i \in \mathcal{P}_D : \ x_{0,i} = 1\}$, $\mathcal{P}_N \triangleq \{\forall i \in \mathcal{P}_N : \ x_{0,i} = 0\}$ and $\mathcal{P} \triangleq \{1,\ldots,\mathrm{N}\}$, respectively,

Vector $\mathbf{x}$ (according to the signal model in chapter 1) can be also described by three sets which enclose all the subjects at the end of the decoding stage. First we define the set of subjects that remain unlabeled by the erasure decoder as

Definition 4.2 (Unlabeled subjects of the population $\mathcal{P}$)**.** $\mathcal{U} \triangleq \{\forall i \in \mathcal{U} : \ x_i = ?\}$.

This set encloses all the indexes of vector $\mathbf{x}$ that remain unlabeled at the end of the decoding stage. Then we can define the set of defective subjects that were labeled by the erasure decoder as

Definition 4.3 (Set of defective subjects of $\mathcal{P}$ that are labeled)**.** $\mathcal{D} \triangleq \mathcal{P}_D \cap \mathcal{U}^c$,

1 The generalisation of all the results in this chapter for $\mathrm{P}(P=1) > \mathrm{P}(P=0)$ and an AND-based encoder $\mathrm{g}_a(\cdot)$ in equation (1.4) is straightforward since we just have to swap '0' and by '1' and '1' and by '0'.

and the set of non-defective subjects of $\mathcal{P}$ that were labeled as

Definition 4.4 (Set of non-defective subjects of $\mathcal{P}$ that are labeled)**.** $\mathcal{N} \triangleq \mathcal{P}_N \cap \mathcal{U}^c$.

Finally we define the complimentary set of $\mathcal{U}$ as $\mathcal{U}^c \triangleq \mathcal{P} \setminus \mathcal{U}$ and the complementary sets of $\mathcal{D}$ and $\mathcal{N}$ as $\mathcal{D}^c \triangleq \mathcal{P}_D \cap \mathcal{U}^c$ (or alternatively $\mathcal{D}^c \triangleq \mathcal{P}_D \setminus \mathcal{D}$) and $\mathcal{N}^c \triangleq \mathcal{P}_N \cap \mathcal{U}^c$ (or alternatively $\mathcal{N}^c \triangleq \mathcal{P}_N \setminus \mathcal{N}$).

Let us remark that we use probabilities to measure the relations between the sets, i.e. $\mathrm{P}(x \in \mathcal{U}) = |\mathcal{U}|/|\mathcal{P}|$ refers to the probability of unlabeled subjects of the population. For the sake of clarity we always omit the reference to 'x', so we write $\mathrm{P}(\mathcal{D}^c)$ to refer to the probability of defective subjects of the population that remain unlabeled at the end of the decoding step and $\mathrm{P}(\mathcal{D}^c \mid \mathcal{U}) = |\mathcal{D}^c|/|\mathcal{U}|$ to denote the fraction of defective subjects among those that remain unlabeled.

Next we define the sets that we require to model the performance of the non-adaptive group testing scheme in the N-asymptotic regime. We define four sets partitioning the messages sent from variable nodes (v) to check nodes (c) in the graph depending on whether (v) is defective or not and whether it has been already labeled or not. Analogously, we also define four additional sets for the messages sent from check nodes (c) to variable nodes (v).

Definition 4.5 ($\mathcal{D}_{c\to v}|\mathcal{P}_D$)**.** *Set of edges that carry a defective message conveyed by a check node to a variable node that represents a defective component.*

Definition 4.6 ($\mathcal{D}^c_{c\to v}|\mathcal{P}_D$)**.** *Set of edges that carry an unlabeled message conveyed by a check node to a variable node that represents a defective component.*

Definition 4.7 ($\mathcal{N}_{c\to v}|\mathcal{P}_N$)**.** *Set of edges that carry a non-defective message conveyed by a check node to a variable node that represents a non-defective component.*

Definition 4.8 ($\mathcal{N}^c_{c\to v}|\mathcal{P}_N$)**.** *Set of edges that carry an unlabeled message conveyed by a check node to a variable node that represents a non-defective component.*

Definition 4.9 ($\mathcal{D}_{v\to c}|\mathcal{P}_D$)**.** *Set of edges that carry a defective message conveyed by a variable node that represents a defective component to a check node.*

Definition 4.10 ($\mathcal{D}^c_{v\to c}|\mathcal{P}_D$)**.** *Set of edges that carry an unlabeled message conveyed by a variable node that represents a defective component to a check node.*

Definition 4.11 ($\mathcal{N}_{v\to c}|\mathcal{P}_N$)**.** *Set of edges that carry a non-defective message conveyed through the edge by a variable node that represents a non-defective component to a check node.*

Definition 4.12 ($\mathcal{N}^c_{v\to c}|\mathcal{P}_N$)**.** *Set of edges that carry an unlabeled message conveyed by a variable node that represents a non-defective component to a check node.*

4.2 Analysis of graph-based group testing encoder

4.2.1 Labeling capability of a non-adaptive group testing encoder

The author in [61] intuitively stated that the OR based codes can only be asymptotically optimum when $\mathrm{P}(\mathcal{P}_D)$ goes to zero, because the encoder of equation (1.5) cannot perform a one to one mapping between the source sequences $\mathbf{x}_0$ that represent realizations of the population and its compressed versions $\mathbf{y}_0$ when $\mathrm{N} > \mathrm{M} > \mathrm{K} > 0$, i.e. that $\mathrm{P}(\mathcal{U})$ is non-vanishing even if N goes to infinity.

Indeed, in section 2.3.3 we showed that a given fraction of the check nodes cannot label any subjects, i.e. the so called Type III check nodes, and hence there is a chance that a variable node is only connected to this kind of check nodes so it cannot be labeled. We will say that subjects represented by variable nodes only connected to the Type III check nodes are *irretrievable*,

Definition 4.13 (Irretrievable subjects)**.** *Consider a sparse matrix* $\mathbf{A}$*, an OR-based encoder, a sequence* $\mathbf{x}_0$ *and* $\mathbf{y}_0 = \mathrm{q}_o(\mathbf{A},\mathbf{x}_0)$*. We say that a subject (variable node) is irretrievable whenever all the* $|\mathcal{I}^v|$ *check nodes in its neighborhood* $\mathcal{I}^v$ *have more than one defective contributor in their neighborhoods, i.e. for all* $j \in \mathcal{I}^v : |\mathcal{I}_j^c \cap \mathcal{P}_d| > 1$*.*

Let us define some sets. Consider a sparse matrix $\mathbf{A}$, an OR-based encoder, a sequence $\mathbf{x}_0$ and $\mathbf{y}_0 = \mathrm{q}_o(\mathbf{A},\mathbf{x}_0)$, then

Definition 4.14 ($\mathcal{U}_{min}$)**.** *The set of irretrievable subjects of the population.*

Definition 4.15 ($\mathcal{D}_{min}$)**.** *The set of retrievable defective subjects of the population.*

Definition 4.16 ($\mathcal{N}_{min}$)**.** *The set of retrievable non-defective subjects of the population.*

Their complementary sets are $\mathcal{N}_{min}^c \triangleq \mathcal{P}_N \setminus \mathcal{N}_{min}$, $\mathcal{D}_{min}^c \triangleq \mathcal{P}_D \setminus \mathcal{D}_{min}$ and $\mathcal{U}_{min}^c \triangleq \mathcal{P} \setminus \mathcal{U}_{min}$.

The set $\mathcal{U}_{min}$ can be employed to obtain a non-achievable lower bound of the minimum amount of subjects that cannot be labeled by any decoder. As stated before, $\mathcal{U}_{min}$ includes the subjects that are connected exclusively to Type III checks, but there are also subjects that cannot be labeled among those connected to Type II checks. Indeed, those defective subjects whose connections to Type II check nodes always include contributions from non-defective subjects in $\mathcal{N}_{min}^c$ cannot be labeled by any decoder[1].

1 Decoders for non-adaptive group testing.

We can easily deduce $\mathrm{P}(\mathcal{N}^c_{min} \mid \mathcal{P}_N)$ and $\mathrm{P}(\mathcal{D}^c_{min} \mid \mathcal{P}_D)$ if we notice that a subject (defective or non-defective) can be retrieved only if at least contributes to one test where all the other contributors are non-defective. Indeed, remark 2.2 states it for the defective and remark 2.1 alternatively also states that a non-defective subject can be labeled only if all the other subjects are non-defective. Thus, all subjects have the same chance of being retrieved by a decoder. Let us enclose this result in a lemma

Lemma 4.1. *The fraction of defective and non-defective subjects in $\mathcal{U}_{min}$ are equal to the fraction of unlabeled subjects of the population*

$$\mathrm{P}\big(\mathcal{U}_{min}\big) = \mathrm{P}\big(\mathcal{D}^c_{min} \,\big|\, \mathcal{P}_D\big) = \mathrm{P}\big(\mathcal{N}^c_{min} \,\big|\, \mathcal{P}_N\big)$$

We can write $\mathrm{P}\big(\mathcal{U}_{min}\big)$ as the probability that a variable node is only connected to Type III check nodes as[1]

$$\mathrm{P}\big(\mathcal{U}_{min}\big) = \sum_{d_v=1}^{+\infty} \mathrm{L}_{d_v} \left(1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(|\{\mathcal{I}^c \setminus x\} \cap \mathcal{P}_D| = 0 \mid |\mathcal{I}^c| = d_c)\right)^{d_v} \tag{4.1}$$

where L_{d_v} is the fraction of variable nodes with degree d_v. In equation (4.1) we averaged the probability among all the possible variable nodes degrees and is the average number of edges per check node

$$\bar{d}_c = \sum_{\forall d} d\mathrm{R}(d)$$

Note that $|\{\mathcal{I}^c \setminus x\} \cap \mathcal{P}_D| = 0$ indicates that all the other subjects of the population indexed in the neighborhood of $\mathcal{I}^c$ are non-defective. This probability is $\mathrm{P}(\mathcal{P}_N)^{d_c - 1}$ when $|\mathcal{I}^c| = d_c$. Then equation (4.1) reduces to

$$\mathrm{P}(\mathcal{U}_{min}) = \sum_{d_v=1}^{+\infty} \mathrm{L}_{d_v} \left(1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(\mathcal{P}_N)^{d_c - 1}\right)^{d_v} \tag{4.2}$$

and we have the average fraction of subjects that cannot be retrieved as a function of the ensemble Γ(L,R) and the source statistics. Let us enclose this result in the lemma

Lemma 4.2. *Consider an OR-based encoder function $\mathrm{q}_o(\cdot)$ and a sparse matrix $\mathbf{A}$ randomly chosen from ensemble Γ(R,L) in the N-asymptotic regime. Consider a population $\mathcal{P}$ with a reduced*

1 I.e. one minus the probability that all the other contributors to a test are non-defective.

and non-vanishing amount of defective subjects characterized by its binary entropy rate $\mathrm{H}(P) > 0$*, then the encoder leaves systematically a non-vanishing fractions of subjects of the population no less than* $\mathrm{P}(\mathcal{U}_{min})$ *irretrievable by any decoder.*

We can deduce also that

Corollary 4.1. *Consider a graph-based group testing encoder in the* N*-asymptotic regime* Γ(R,L)*. The entropy rate of the r.v.* P *that models the population, i.e.* $\mathrm{H}(P)$*, is equal to the entropy rate of a source modeling the subjects in* $\mathcal{U}_{min}$ *and in* $\mathcal{U}_{min}^{c}$*.*

Corollary 4.1 holds because due to lemma 4.1 it holds that $\mathrm{P}\big(\mathcal{P}_N\big) = \mathrm{P}\big(\mathcal{P}_N \;\big|\; \mathcal{U}_{min}\big) = \mathrm{P}\big(\mathcal{P}_N \;\big|\; \mathcal{U}_{min}^{c}\big)$.

4.3 Limitations of the erasure decoder

The aim of this section is to analyze the performance of the erasure decoding algorithm for non-adaptive group testing on a sparse graph. Here we give closed-form expressions to characterize the performance of an encoder-decoder scheme for group testing formed by the encoder given in equation (1.5) and the erasure decoder presented in section 2.3.2, from now on referred to as $\mathrm{g}_e(\cdot)$.

In this section we give closed-form expressions to characterize the performance of the encoder-decoder scheme for a graph picked at random from a ensemble Γ(L,R) in the N-asymptotic regime and deduce some properties of the non-adaptive group testing scheme.

4.3.1 Analysis of the graph-based encoder and erasure decoder

The erasure decoder attempts to infer the values of the components of $\mathbf{x}_0$ in a process described by algorithm 2.1 in section 2.3.2. The inference process described there can be regarded as a decoding algorithm that performs two iterations: in a first iteration the non-defective subjects are inferred from the non-defective tests of $\mathbf{y}_0$. After that, in a second iteration, the information gathered by the variable nodes is forwarded to the check nodes in their respective neighborhoods which, in their turn, perform an attempt to identify the defective subjects and forward this information to the variable nodes identified as defective.

We aim our analysis to find

$$\mathrm{P}(\mathcal{U}) = \mathrm{P}(\mathcal{D}^c) + \mathrm{P}(\mathcal{N}^c) \tag{4.3}$$

$$\mathrm{P}(\mathcal{U}) = \mathrm{P}\big(\mathcal{D}^c \;\big|\; \mathcal{P}_D\big)\,\mathrm{P}(\mathcal{P}_D) + \mathrm{P}\big(\mathcal{N}^c \;\big|\; \mathcal{P}_N\big)\,\mathrm{P}(\mathcal{P}_N) \tag{4.4}$$

The equations that characterize the inference of the non-defective subjects can we easily deduced because, as we said in section 4.2.1 , $\mathrm{P}(\mathcal{N}^c_{min} \mid \mathcal{P}_N)$ coincides with $\mathrm{P}(\mathcal{N}^c \mid \mathcal{P}_N)$ since both events arise from remark 2.1. Let us enclose this result in a corollary

Corollary 4.2. $\mathrm{P}(\mathcal{U}_{min}) = \mathrm{P}(\mathcal{N}^c \mid \mathcal{P}_N)$

holds due to lemma 4.1. We focus on modelling the second iteration of the erasure decoder, since the fraction of non-defective subjects that cannot be inferred is the one given in equation (4.2), as stated by corollary 4.2, and depends only on the ensemble the graph belongs to and the statistics of the population $\mathcal{P}$. So we focus on deducing a closed-form expression for the term $\mathrm{P}\big(\mathcal{D}^c \,\big|\, \mathcal{P}_D\big)$, also as a function of the graph ensemble and the source statistics.

We can write the probability that an arbitrary i-th variable node representing a defective subject remains unlabeled at the end of the process as

$$\mathrm{P}\big(x_i \in \mathcal{D}^c \,\big|\, x_i \in \mathcal{P}_D\big) = \prod_{j \in \mathcal{I}_i^v} \mathrm{P}\big(\mathcal{D}^c_{c_j \to v_i} \,\big|\, \mathcal{P}_D\big). \tag{4.5}$$

This equation represents the existing dependence between the value of the variable node and the messages that the check nodes in its neighborhood send to it. It reflects that the variable node remains unlabeled in case that all check nodes in $\mathcal{I}_i^v$ would be unable to infer the label of the i-th variable node. Averaging over the ensemble the latter expression becomes

$$\mathrm{P}\big(\mathcal{D}^c \,\big|\, \mathcal{P}_D\big) = \sum_{d_v=1}^{+\infty} \mathrm{L}_{d_v}\, \mathrm{P}\big(\mathcal{D}^c_{c \to v} \,\big|\, \mathcal{P}_D\big)^{d_v} \tag{4.6}$$

We can deduce the term $\mathrm{P}\big(\mathcal{D}^c_{c \to v} \,\big|\, \mathcal{P}_D\big)$ as follows

$$\mathrm{P}\big(\mathcal{D}^c_{c \to v} \,\big|\, \mathcal{P}_D\big) = 1 - \mathrm{P}\big(\mathcal{D}_{c \to v} \,\big|\, \mathcal{P}_D\big) \tag{4.7}$$

then

$$\begin{aligned} \mathrm{P}\big(\mathcal{D}_{c \to v} \,\big|\, \mathcal{P}_D\big) &= \frac{\sum_{d_c=1}^{+\infty} \mathrm{R}_{d_c}\, \mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \,\big|\, |\mathcal{I}^c| = d_c\big)}{\sum_{d_c=1}^{+\infty} d_c \mathrm{R}_{d_c}\, \mathrm{P}(\mathcal{P}_D)} \\ &= \sum_{d_c=1}^{+\infty} \frac{\mathrm{R}_{d_c}}{\bar{d}_c} \frac{\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \,\big|\, |\mathcal{I}^c| = d_c\big)}{\mathrm{P}(\mathcal{P}_D)} \end{aligned} \tag{4.8}$$

where the event $|\mathcal{I}^c \cap \mathcal{P}_D| = 1$ indicates that only a defective subject contributed to generate the test of the check node and the event $|\mathcal{I}^c \cap \mathcal{N}| = d_c - 1$ indicates that all the other subjects that

contributed to the test were labeled in the previous iteration. Then we can write

$$\begin{aligned}&\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \;\big|\; |\mathcal{I}^c| = d_c\big) = \\ &= \mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \;\big|\; |\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c| = d_c\big)\, \mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{P}_D| = 1 \;\big|\; |\mathcal{I}^c| = d_c\big)\end{aligned} \tag{4.9}$$

where the term $\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{P}_D| = 1 \;\big|\; |\mathcal{I}^c| = d_c\big)$ is given in equation (2.4) and the term $\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \;\big|\; |\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c| = d_c\big)$ can be written as

$$\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \;\big|\; |\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c| = d_c\big) = \frac{term_1 - term_2}{term_3 - term_2} \tag{4.10}$$

where the $term_1$ is the average number of edges connected to variable nodes in $\mathcal{N}$ (equation 4.11), $term_2$ represents the average number of edges of the graph connected to check nodes so as $|\mathcal{I}^c \cap \mathcal{P}_D| = 0$ (equation 4.12) and $term_3$ is the average number of edges connected to variable nodes in $\mathcal{P}_N$ (equation 4.13)

$$term_1 = \sum_{d_c=1}^{+\infty} d_c \mathrm{R}_{d_c} \big(1 - \mathrm{P}\big(\mathcal{N}^c \;\big|\; \mathcal{P}_N\big)\big)\, \mathrm{P}(\mathcal{P}_N) = \bar{d}_c \big(1 - \mathrm{P}\big(\mathcal{N}^c \;\big|\; \mathcal{P}_N\big)\big)\, \mathrm{P}(\mathcal{P}_N) \tag{4.11}$$

$$term_2 = \sum_{d_c=1}^{+\infty} d_c \mathrm{R}_{d_c}\, \mathrm{P}(|\mathcal{I}^c \cap \mathcal{P}_D| = 0) = \sum_{d_c=1}^{+\infty} d_c \mathrm{R}_{d_c}\, \mathrm{P}(\mathcal{P}_N)^{d_c} \tag{4.12}$$

$$term_3 = \sum_{d_c=1}^{+\infty} d_c \mathrm{R}_{d_c}\, \mathrm{P}(\mathcal{P}_N) = \bar{d}_c\, \mathrm{P}(\mathcal{P}_N) \tag{4.13}$$

All the terms in equations (4.11, 4.12, 4.13) have been either already deduced[1] or depend on the the graph ensemble and the statistics of the population source. Combining (4.11, 4.12, 4.13) into (4.10) and operating we obtain

$$\begin{aligned}&\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{N}| = d_c - 1 \;\big|\; |\mathcal{I}^c \cap \mathcal{P}_D| = 1, |\mathcal{I}^c| = d_c\big) = \\ &= \frac{1 - \mathrm{P}\big(\mathcal{N}^c \;\big|\; \mathcal{P}_N\big) - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c}\, \mathrm{P}(\mathcal{P}_N)^{d_c - 1}}{1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c}\, \mathrm{P}(\mathcal{P}_N)^{d_c - 1}} = \\ &= 1 - \frac{\mathrm{P}\big(\mathcal{N}^c \;\big|\; \mathcal{P}_N\big)}{1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c}\, \mathrm{P}(\mathcal{P}_N)^{d_c - 1}}\end{aligned} \tag{4.14}$$

1 the term $\mathrm{P}\big(\mathcal{N}^c \;\big|\; \mathcal{P}_N\big)$ is equal to $\mathrm{P}(\mathcal{U}_{min})$ in equation (4.2) due to lemma 4.1.

and, then we can compute the term $\mathrm{P}\big(\mathcal{D}_{c\to v} \mid \mathcal{P}_D\big)$ in equation 4.8 as

$$\mathrm{P}\big(\mathcal{D}_{c\to v} \mid \mathcal{P}_D\big) = \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(\mathcal{P}_N)^{d_c-1} \left(1 - \frac{\mathrm{P}\big(\mathcal{N}^c \mid \mathcal{P}_N\big)}{1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(\mathcal{P}_N)^{d_c-1}}\right) \tag{4.15}$$

Finally we can substitute equation (4.15) in equation (4.7) and this result into (4.6) to obtain

$$\mathrm{P}\big(\mathcal{D}^c \mid \mathcal{P}_D\big) = \sum_{d_v=1}^{+\infty} \mathrm{L}_{d_v} \left(1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(\mathcal{P}_N)^{d_c-1} \left(1 - \frac{\mathrm{P}\big(\mathcal{N}^c \mid \mathcal{P}_N\big)}{1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(\mathcal{P}_N)^{d_c-1}}\right)\right)^{d_v} \tag{4.16}$$

and then plug the resulting term and $\mathrm{P}\big(\mathcal{N}^c \mid \mathcal{P}_N\big)$ obtained as in equation (4.2) into equation (4.4) to obtain the average fraction of subjects that remain unlabeled after the decoding step $\mathrm{P}\big(\mathcal{U}\big)$ as a function of the ensemble of graphs Γ(R,L) and the source statistics.

4.3.2 Results

Here we compare the encoder limits obtained in section 4.2.1 with the limits of the encoder-decoder scheme obtained in subsection 4.3.1.

To compare the performance of the encoder-decoder scheme and the encoder limit we have to compare both the pair of probabilities $\mathrm{P}(\mathcal{N}^c_{min})$ and $\mathrm{P}(\mathcal{N}^c)$ with their counterparts $\mathrm{P}(\mathcal{D}^c_{min})$ and $\mathrm{P}(\mathcal{D}^c)$. On the one hand, $\mathrm{P}(\mathcal{N}^c)$ and $\mathrm{P}(\mathcal{N}^c_{min})$ are equal as stated by corollary 4.2. On the other side, it does not happen the same with the other pair because, simply by comparing equations (4.16) and equation (4.2)[1]we can see that there is an extra factor into brackets that

$$\left(1 - \frac{\mathrm{P}\big(\mathcal{N}^c \mid \mathcal{P}_N\big)}{1 - \sum_{d_c=1}^{+\infty} \frac{d_c \mathrm{R}_{d_c}}{\bar{d}_c} \mathrm{P}(\mathcal{P}_N)^{d_c-1}}\right) < 1$$

and hence $\mathrm{P}(\mathcal{D}^c_{min}) > \mathrm{P}(\mathcal{D}^c)$. Let us enclose this result in a lemma

Lemma 4.3. *Consider a population $\mathcal{P}$ with $\mathrm{P}(\mathcal{P}_N) > \mathrm{P}(\mathcal{P}_D) > 0$ and a graph from a ensemble $\Gamma\big(\mathrm{L},\mathrm{R}\big)$, an OR-based encoder as described by equation (1.5) and the erasure decoder. In the* N*-asymptotic regime it holds that*

$$\mathrm{P}(\mathcal{U}_{min}) < \mathrm{P}(\mathcal{U}), \quad \mathrm{P}(\mathcal{D}^c_{min}) < \mathrm{P}(\mathcal{D}^c), \quad \mathrm{P}(\mathcal{N}^c_{min}) = \mathrm{P}(\mathcal{N}^c)$$

1 Note that due lemma 4.1 it holds that $\mathrm{P}\big(\mathcal{D}^c_{min} \mid \mathcal{P}_D\big) = \mathrm{P}\big(\mathcal{U}_{min}\big)$.

Next lemma arises from lemma 4.3:

Lemma 4.4. *Consider a population* $\mathcal{P}$ *with* $\mathrm{P}(\mathcal{P}_N) > \mathrm{P}(\mathcal{P}_D) > 0$ *and a graph from a ensemble* $\Gamma\big(\mathrm{L},\mathrm{R}\big)$*, an OR-based encoder as described by equation (1.5) and the erasure decoder. In the* N*-asymptotic regime it holds that*

$$\mathrm{H}(P) = \mathrm{H}(\mathcal{U}_{min}) \leq \mathrm{H}(\mathcal{U})$$

where $\mathrm{H}(P)$ *is the binary entropy per subject of the population,* $\mathrm{H}(\mathcal{U}_{min})$ *is the binary entropy per non-retrievable subject of the population and* $\mathrm{H}(\mathcal{U})$ *is binary entropy per unlabeled subject of the population.*

The latter lemma follows from lemmas 4.1 and 4.3.

Then, if follows that

Lemma 4.5. *Consider a graph-based group testing encoder in the* N*-asymptotic regime* $\Gamma(\mathrm{R},\mathrm{L})$ *and a population* $\mathcal{P}$ *with* $\mathrm{P}(\mathcal{P}_N) > \mathrm{P}(\mathcal{P}_D) > 0$ *characterized by its entropy rate* $\mathrm{H}(P)$*. The encoder performs a mapping between the approximately* $2^{\mathrm{N}\,\mathrm{H}(P)}$ *sequences in the typical set of the population* $\mathcal{P}$*, into strictly less than* $2^{\mathrm{N}\left(1-\mathrm{P}(\mathcal{U}_{min})\right)\mathrm{H}(P)}$ *different sequences of tests.*

Proof. Lemma 4.5 holds because due to lemmas 4.4 and 4.3

$$2^{\mathrm{N}\,\mathrm{H}(U)\,\mathrm{P}(\mathcal{U})} > 2^{\mathrm{N}\,\mathrm{H}(P)\,\mathrm{P}(\mathcal{U}_{min})} \tag{4.17}$$

where

$$\mathrm{H}(P) = -\,\mathrm{P}\big(\mathcal{N}^c_{min} \mid \mathcal{U}_{min}\big)\log_2(\mathrm{P}\big(\mathcal{N}^c_{min} \mid \mathcal{U}_{min}\big)) - \mathrm{P}\big(\mathcal{D}^c_{min} \mid \mathcal{U}_{min}\big)\log_2(\mathrm{P}\big(\mathcal{D}^c_{min} \mid \mathcal{U}_{min}\big))$$

due to corollary 4.1 and

$$\mathrm{H}(U) = -\,\mathrm{P}\big(\mathcal{N}^c \mid \mathcal{U}\big)\log_2(\mathrm{P}\big(\mathcal{N}^c \mid \mathcal{U}\big)) - \mathrm{P}\big(\mathcal{D}^c \mid \mathcal{U}\big)\log_2(\mathrm{P}\big(\mathcal{D}^c \mid \mathcal{U}\big))$$

and if we swap the terms in equation (4.17) and multiply by $2^{\mathrm{N}\,\mathrm{H}(P)}$ both sides, it follows that

$$2^{\mathrm{N}(\mathrm{H}(P)-\mathrm{P}(\mathcal{U})\,\mathrm{H}(U))} < 2^{\mathrm{N}\left(1-\mathrm{P}(\mathcal{U}_{min})\right)\mathrm{H}(P)}$$

□

The reader should note that the author in [157] states that within the noiseless non-adaptive GT setup and with a non-vanishing fraction of defective subjects there exists a non-zero threshold p^*, so as for $\mathrm{P}(\mathcal{P}_D) < p^*$ "an arbitrarily accurate estimation is possible" whereas $\mathrm{P}(\mathcal{P}_D) > p^*$ "no estimator that can achieve an arbitrarily small error probability exists in the asymptotic limit". This result obviously contradicts the results obtained in section 4.2 and section 4.3 where we show that no estimator that can achieve an arbitrarily small error probability exists in the N-asymptotic limit for $\mathrm{P}(\mathcal{P}_D) > 0$ since the encoder is non-injective, as stated by lemma 4.5 and this result follows from lemma 4.2 that states that a GT encoder leaves systematically a non-vanishing fraction of subjects of the population unable to be labeled by any decoder.

We can illustrate with an example that there is some conceptual problem in the derivation of the thresholds [157] since it is quite easy to show that it is impossible to infer the label of several defective and non-defective subjects of the population from a vector of test generated by a non-adaptive group testing encoder when $\mathrm{K} = \mathrm{O}(\mathrm{N})$ and N is an arbitrarily large number.

The author states that the reconstruction threshold p^* for a regular bipartite graph with $d_v = 2$ and $d_c = 4$ is sandwiched by $0.92763 < p^* < 0.097350$. Figure 4.1 depicts a sub-graph of an ensemble with $d_c = 4$ and $d_v = 2$ where circles and squares represent variable and check nodes, respectively.

The black circles represent labeled defective subjects and the white circles represent to labeled non-defective subjects whereas the circle at the top of the graph represents an unlabeled variable node.

In a general case, the non-defective variable nodes at the bottom of the figure are labeled with an arbitrary non-zero and non-vanishing probability $\mathrm{P}(\mathcal{N})$[1].

So, due to remark 2.2 each check node connected to the variable nodes at the bottom of the figure can label the remaining variable node as defective and this fact happens with probability

$$\mathrm{P}(\mathcal{D}) = \mathrm{P}(\mathcal{P}_D)\,\mathrm{P}(\mathcal{N})^3.$$

In this situation, it is straightforward that the real label of the variable node connected to the top layer of check nodes cannot infer it, irrespective of its value. So, with a certain probability (in this case $(\mathrm{P}(\mathcal{D})^3)^2$) the label of the variable node at the top of the figure cannot be inferred. Note that this probability does not vanish when N increases since the number of non-defective subjects is $\mathrm{N}\,\mathrm{P}(\mathcal{P}_N)$.

1 If we assume that they were labeled by a genie decoder this probability would be 1 or a number that depends of $\mathrm{P}(\mathcal{P}_N)$ if we use the erasure decoder.

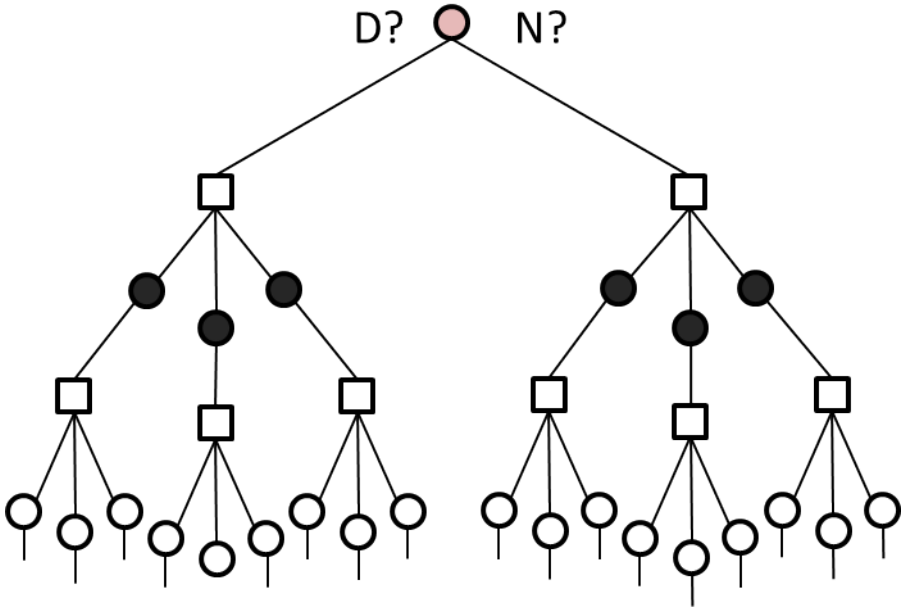


Figure 4.1: Example of graph with $d_c = 4$ and $d_v = 2$. Black/white circles represent defective/non-defective variable nodes. Squares represent check nodes.

So, in general, in the N-asymptotic regime this specific event happens with a non-vanishing probability in a bipartite graph with $d_v = 2$ and $d_c = 4$ and it is repeated an arbitrarily large number of times that depends on N. Furthermore the fraction of defective and non-defective subjects whose labels cannot be inferred due to this event is equal to $\mathrm{P}(\mathcal{P}_D)$ and $\mathrm{P}(\mathcal{P}_N)$, respectively, since the labels of the subjects are generated independently and, so if we declare all the unlabeled subjects as defective we would commit a non-vanishing probability of error, e.g. in our example $(\mathrm{P}(\mathcal{D})^3)^2\,\mathrm{P}(\mathcal{P}_N)$, Similarly, if we declare all the subjects as non-defective we will commit also a non-vanishing error $(\mathrm{P}(\mathcal{D})^3)^2\,\mathrm{P}(\mathcal{P}_D)$.

4.4 Progressive encoding for binary source compression

In this section we introduce a fixed rate adaptive source coding scheme that we presented in [116], referred to as Progressive encoding for hybrid LDPrC-LDPC codes. It is a modification of the original hybrid LDPrC-LDPC coding scheme reviewed in section 2.2.3 that can be employed to obtain a family of codes with better performance while keeping the computational complexity low. It is based on the generation of the encoded bits in a sequential manner by stages rather than all at the same time.

The proposed scheme allows to compress very low entropy sources with a rate losses around 20 %. While this figure might seem poor taking into account that there exist optimum source coding schemes in the literature (e.g. entropy coding), this scheme departs from these optimum

procedures in that the complexity is moved from the encoder to the decoder. LDPC [95] and turbo codes [62, 63] have been proposed in the past to achieve this goal, but the performance of linear codes for compression experiences significant degradation when the source entropy decreases (the smaller the entropy the higher the relative gap between the code performance and the theoretical limit).

4.4.1 System setup

We consider the problem of almost loss-less source coding of an asymmetric memory-less binary Bernoulli source with $\mathrm{P}(X = 1)) > \mathrm{P}(X = 0)$. We consider fixed-length block source codes with the AND-based encoder as in equation 1.4, where a sequence of N information bits, $\mathbf{x}_0$, is compressed into a codeword of $\mathrm{M} < \mathrm{N}$ bits, so that a code with compression rate $r = \mathrm{M}/\mathrm{N}$ is obtained.

4.4.2 Encoding procedure

The encoding procedure for the hybrid codes is modified so that it is performed in successive stages. At each stage a piece of the codeword is obtained. If we denote by $\mathbf{y}_i$ the piece of the compressed sequence generated at the i-th stage and by m the number of stages, then

$$\mathbf{y}_0 = \left[\mathbf{y}_1, \ldots, \mathbf{y}_\mathrm{m}\right] \tag{4.18}$$

The basic idea is to use the erasure decoder described in section 2.3.2 at the encoder side to identify at the earliest possible stage the maximum number of components of vector $\mathbf{x}_0$ from a reduced set of components of the compressed sequence $\mathbf{y}_0$ and to devote the remaining components of the sequence $\mathbf{y}_0$ to compress those components that remain as '?' after the erasure decoder at the end of each stage.

The procedure is as follows: in the first stage, all components of vector $\mathbf{x}_0$ contribute to generate $\mathbf{y}_1$.

$$\mathbf{y}_1 = \mathrm{q}_a(\mathbf{A}_1, \mathbf{x}_0) \tag{4.19}$$

where $\mathbf{A}_1$ is a random matrix of the ensemble of matrices of size $\mathrm{M}_1 \times \mathrm{N}$ that has $d_v = 1$ and check node degree $d_c^{(1)}$ and hence $\mathrm{M}_1 = \mathrm{N}/d_c^{(1)}$. Afterwards, the encoder tries to recover the components of vector $\mathbf{x}_0$ from $\mathbf{y}_1$ employing the erasure decoder. Denote as $(1 - f_1)\mathrm{N}$ the amount of source bits that can be recovered, and as $f_1\mathrm{N}$ the bits that remain as '?'.

In the second stage, the f_1N bits that remain as '?' are encoded again

$$\mathbf{y}_2 = \mathrm{q}_a(\mathbf{A}_2, \mathbf{x}_0) \tag{4.20}$$

with an AND-based encoder but, as opposed to $\mathbf{A}_1$, in this case the matrix $\mathbf{A}_2$ has a structure $\mathbf{A}_2 = [\mathbf{0}\ \mathbf{B}_2]\Pi_2$ where $\mathbf{0}_2$ is a zero matrix of size $\mathrm{M}_2 \times (1 - f_1)\mathrm{N}$, $\mathbf{B}_2$ is a random matrix of the ensemble of matrices of size $\mathrm{M}_2 \times (f_1\mathrm{N})$ represented by a regular graph with $d_v = 1$ and check node degree $d_c^{(2)}$ (so $\mathrm{M}_2 = f_1\mathrm{N}/d_c^{(2)}$) and Π_2 is a permutation matrix of size $\mathrm{N} \times \mathrm{N}$ that rearranges the rows of $\mathbf{B}_2$ so that the non-zero rows are mapped to the components of vector $\mathbf{x}_0$ that remain as '?' after the first stage. So, the compressed sequence generated up to the second encoding stage is

$$\mathbf{y} = \left[\mathbf{y}_1\ \mathbf{y}_2\right] = \mathrm{q}_a([\mathbf{A}_1\ \mathbf{A}_2],\mathbf{x}_0) \tag{4.21}$$

Note that the check and variable node degree profiles of the overall graph $[\mathbf{A}_1\ \mathbf{A}_2]$ are irregular now.

Next, the encoder tries to recover the components of vector $\mathbf{x}_0$ that remained as '?' employing the 'erasure decoder' $\mathbf{x} = \mathrm{g}_e([\mathbf{A}_1\ \mathbf{A}_2],\mathbf{y})$. Denote as f_2N the bits that remain as '?' after decoding the second stage. These bits are further encoded in stage 3 with a matrix $\mathbf{B}_3$ and check nodes of degree $d_c^{(3)}$ so $\mathrm{M}_3 = f_2\mathrm{N}/d_c^{(3)}$. Afterwards the 'erasure decoder' is applied again and the procedure follows with as many stages as desired. Hence, for I stages

$$\mathrm{M} = \sum_{i=1}^{I} \mathrm{M}_i = \sum_{i=1}^{I} f_i\mathrm{N}/d_c^{(i)} \tag{4.22}$$

and the total code rate for all stages is

$$r = \mathrm{M}/\mathrm{N} = \sum_{i=1}^{I} f_i/d_c^{(i)} \tag{4.23}$$

and f_IN bits remain as '?' at the end of this procedure.

Note that after each stage the fraction of components that remain as '?' and need to be further processed decreases ($1 > f_1 > f_2 > f_3 > \ldots f_I$). Furthermore, as most of the components being identified in early encoding stages are '1"s, the entropy of the components that remaining to be labeled increases. This procedure can be continued until no bits remain as '?' or the entropy is very close to one.

Alternatively, this procedure can also be stopped after any number of stages. In this case the bits remaining as '?' can be encoded with a classic LDPC code of smaller size. Note that

although components of vector $\mathbf{y}$ are generated in several stages, the whole encoding process can be regarded as a single hybrid LDPrC-LDPC code with irregular degree profiles whose matrix depends on the source sequence to be compressed. This dependency appears through the permutation matrices $\Pi_2, \ldots, \Pi_I$, but the random matrices $\mathbf{A}_1, \mathbf{B}_2 \ldots \mathbf{B}_I$ belong to ensemble of matrices with fixed dimensions and so the resulting code scheme has fixed rate.

It is important to remark that the same 'erasure decoding' process can be employed at the encoder and the decoder, so at the decoder side one can recover the permutation matrices $\Pi_2, \ldots, \Pi_I$ that were employed in the encoder and reconstruct the graph. At the receiver side, once this the graph has been obtained the fully-fledged decoder described in [106] for hybrid LDPrC-LDPC codes (i.e. the decoder that exploits the knowledge of source entropy and exchanges soft messages) can be applied to get the best performance.

4.4.3 Analysis

The improved performance of the proposed procedure follows from the fact that those components of vector $\mathbf{x}_0$ that are more difficult to recover are also the most protected by the compression code, whereas those that can be easily recovered have smaller degree.

The simulation results in section 4.4.4 indicate that the proposed procedure provides a low complexity method to compress very low entropy sources. The proposed scheme overcomes the limitations of the linear codes in this scenario by mapping this problem into the generation of a set of coded bits by AND operators of high degree and the compression of a remaining sequence of bits of higher entropy, which can be efficiently encoded with a state-of-the-art LDPC code. For example, in section 4.4.4 we show that a source with entropy 0.05 can be compressed using four non-linear stages and an LDPC code of rate 0.5. Thus, the problem of compressing a long source of low entropy is mapped into that of compressing a short sequence of higher entropy. From the code design point of view the only drawback of this procedure is that the LDPC code operates with a smaller codeword length (its input block length is f_IN rather than N), so a careful design of its parity check matrix must be made to guarantee good performance.

Although the encoding procedure described in section 4.4.2 results in a code with variable node degrees i for those bits retrieved in the i-th stage, the generalization of the proposed procedure to obtain an overall hybrid LDPrC-LDPC codes with more irregular degree profiles is straightforward. In section 4.4.4 we show simulations where the check node degree of each stage $d_c^{(i)}$ were selected as the ones that made the probability of check nodes of Type I equal to 0.5, i.e. $d_c^{(i)} = -\log_2^{-1}\big(\mathrm{P}(X_i = 1)\big)$[1] or its closest integer, being $\mathrm{P}(X_i = 1)$ the probability of '1' at the

1 We discuss further the topic of check node degree profile selection with closed-form expressions in chapters 5 and 8. See appendix 5.A for further details on this specific check node degree distribution.

input of the i-th stage. This criterion leads to an encoded sequence with equally likely 0s and 1s (which is a desirable feature in a compressed sequence). Proceeding in this way, the number of coded bits generated at each stage keeps approximately constant ($\mathrm{M}_1 \approx \ldots \approx \mathrm{M}_I$), and on average at least half the variable nodes are determined at each stage.

The fraction of source bits that remain to be processed after each encoding stage (i.e. the values of $f_1,\ldots,f_I$) depends on sequence $\mathbf{x}_0$, and it has a variance that is larger for latter encoding stages and decreases as N increases. Hence, the progressive encoding procedure is most effective when long codewords are considered. A procedure must be proposed to tackle with this variable length without requiring the generation of a new encoding matrix for every sequence to be encoded.

In next section we give simulation results where the check node degrees $d_c^{(i)}$ of each matrix $\mathbf{B}_2,\ldots,\mathbf{B}_I$ were originally designed according to the average length of the sequences at the input of the corresponding stage. Then these matrices where extended adding additional rows corresponding to a number Δ_i of permutation matrices of size n_i, so the degree of the check nodes was increased to $d_c^{(i)}+\Delta_i$ and the number of bits entering the i-th encoding stage was increased from $\mathrm{M}_i d_c^{(i)} = f_{i-1}\mathrm{N}$ to $\mathrm{M}_i(d_c^{(i)}+\Delta_i)$. In those source sequences where some of these bits (usually the $\mathrm{M}_i\Delta_i$ first ones) have been recovered in previous stages, no further encoding is necessary and the source can be replaced at the input of the product node by '1', thereby reducing the effective degree of the product node and getting it closer to the original value of $d_c^{(i)}$. In those source sequences when the sequence length at the input of the i-th stage exceeds $\mathrm{M}_i(d_c^{(i)}+\Delta_i)$ an unrecoverable error will occur.

In the final linear coding stage, a similar procedure must be proposed to cope with the variable input length. In the simulations in section 4.4.4 this issue was approached by increasing the LDPC codeword length while keeping the code rate fixed.

Regarding implementation complexity, note that the operation at the encoder is very simple (only LDPrC-LDPC encoding and erasure decoding are required). It is also important to remark that the data-dependency is only introduced through the permutation matrices Π_i which act over the information bits. Hence, the conventional LDPrC-LDPC decoder presented in [105] can be employed at the receiver by introducing data-dependent interleavers. Note also that the size of sub-matrix $\mathbf{B}_i$ is reduced at every stage, and this fact can be also exploited to reduce decoder complexity.

4.4.4 Simulation results

In this section we assess the performance of the proposed compression procedure to compress two sources with entropy 0.091 and 0.05. Here we simulate a compression scheme where the

encoder employs the progressive encoding procedure to generate the compressed sequence by stages. Afterwards, the encoder conveys the compressed sequence to the decoder. Then, the encoder uses the erasure decoder to obtain partial decoding results to build the same graph that at the decoder side. Finally, the decoder attempts to reconstruct the original sequence employing BP-based message passing algorithms in appendix 2.A, once the decoder has inferred the graph connections.

For the compression of both sources four stages of coded bits are generated by AND-based check nodes and the bits that remained as '?' after these stages were encoded with a linear code.

In the case of source entropy 0.091 a block of 103250 bits was compressed into 11950 coded bits, so a code rate of 0.1157 was obtained. After MC simulation of 16600 codewords, 9 wrong codewords were obtained and the average BER was $1.53\text{x}10^{-6}$. The parameters for the non-linear stages are listed in Table 4.1. As indicated there, an average of 6670 bits remained to be identified after these four stages, and their average entropy was 0.70. These bits where then compressed with an LDPC code of rate 0.5 obtained from [95, 131]. Note that the code rate is smaller than the entropy of these bits. Operation in this regime was possible because the LDPC decoder did not operate alone: it was assisted by the non-linear stages, since the optimum decoder for the hybrid LDPC-LDPrC code was employed. In order to take into account the variability in the number of bits that remain undetermined at the end of the fourth stage the number of parity checks was increased from $f_4\text{N} = 6670$ to 9900, maintaining constant the total code rate. This performance compares favourably with that obtained when a single LDPC code is employed for the same task: in [95] an LDPC code was optimized to compress a source with entropy 0.091 and the minimum rate required for it was 0.125.

In the case of source entropy 0.05 a block of 1230000 bits were compressed into 85500 coded bits, so a code rate of 0.064 was obtained. After Monte Carlo simulation of 2600 codewords 1 wrong codeword was obtained and the average BER was $2.2\text{x}10^{-6}$. The parameters for the non-linear stages are listed in table 4.2. In this case a linear code of rate 0.425 was employed [137].

Table 4.1: Code parameters for source with entropy H = 0.091 and N= 103250. Rate = 0.1157

		NL Stage 1	NL Stage 2	NL Stage 3	NL Stage 4	LDPC
Check node degree $d_c^{(i)} + \Delta_i$		59	29+5	14+5	7+5	
Bits to be encoded at that stage	$\text{P}(X_i = 1)$	0.9888	0.9767	0.9533	0.9074	0.8199
	Entropy	0.091	0.1596	0.2720	0.4449	0.6799
	$\text{M}_i d_c^{(i)} + \text{M}_i \Delta_i$	103250	59500	33250	21000	9900
Number of coded bits (M_i)		1750	1750	1750	1750	4950
Code rate contribution (M_i/N)		0.0169	0.0169	0.0169	0.0169	0.0479
Unlabeled bits after i ($f_i\text{N}$)		51379	25688	12968	6670	

Table 4.2: Code parameters for source with entropy H = 0.05 and N= 1230000. Rate = 0.064

		NL Stage 1	NL Stage 2	NL Stage 3	NL Stage 4	LDPC
Check node degree $d_c^{(i)} + \Delta_i$		123	61+7	30+7	15+7	
Bits to be encoded at that stage	$P(X_i = 1)$	0.9944	0.9775	0.9775	0.9552	0.9115
	Entropy	0.05	0.0888	0.1550	0.2637	0.4313
	$M_i d_c^{(i)} + M_i \Delta_i$	$123 \cdot 10^4$	$68 \cdot 10^4$	$37 \cdot 10^4$	$22 \cdot 10^4$	91100
Number of coded bits (M_i)		10000	10000	10000	10000	38781
Code rate contribution (M_i/N)		0.0081	0.0081	0.0081	0.0081	0.0315
Unlabeled bits after i (f_iN)		613477	306614	153888	77893	

CHAPTER 5

Adaptive sampling for lossless group testing

In this chapter we propose an adaptive scheme for noiseless group testing aimed at building a data-dependent measurement function $\mathbf{A}$ in stages by means of an adaptive process alternating testing and decoding stages where after each decoding stage a new piece of the measurement matrix $\mathbf{A}_i$ is designed exploiting the information available at the output of the decoder at the end of each stage. As in the other adaptive scheme presented in section 4.4, the addition of a new stage of tests reduces the amount of subjects that remain to be labeled after each stage with a given probability. Regretfully the scheme presented in section 4.4 cannot achieve lossless reconstruction condition in the finite size case due to the fact that it works with fixed-size matrices that are the same for all source realizations.

We proposed in [119] an alternative adaptive sampling algorithm where tests are also generated by stages where the matrix of each stage is chosen at random from an ensemble Γ(L,R) and its structure is designed employing the information available at the decoder output at the end of the previous stage. In this design, the number of rows of the resulting measurement matrix depends on the signal realization and furthermore, the amount of stages depends on the rules to choose the ensemble. The structure of the proposed scheme is based on Luby-transform codes [96, 99], a rateless linear coding scheme reviewed in chapter 2.

The resulting scheme can achieve lossless reconstruction of any source sequence $\mathbf{x}_0$ and, furthermore, can be applied to the group testing problem since it relays exclusively on OR-based check nodes, whereas the adaptive scheme presented in 4.4.1 could not, because it was based on hybrid LDPrC-LDPC codes.

This chapter is divided as follows. First in section 5.1 we describe the variable rate adaptive GT scheme that we presented in [119] and in section 5.2 we enclose simulation results that describe its performance in several scenarios and compare it with other adaptive and non-adaptive schemes.

5.1 Scheme overview

This section is devoted to introduce the sequential process that the algorithm follows to both design and to construct matrix $\mathbf{A}$ and identify the defective subjects in the population.

Here in section 5.1.1 we introduce the sequential procedure that the algorithm follows whereas in section 5.1.2 we describe the novel approach to design and construct matrix $\mathbf{A}$ exploiting the tests results obtained in previous stages.

5.1.1 Sequential multi-stage process

In this sequential adaptive group testing scheme tests are generated by stages, as many as required to label all the subjects in the population. A generic i-th encoding stage is composed of three steps:

(1) *A design and construction step*: The check node degree profile R_i is decided and matrix $\mathbf{A}_i$ is constructed as a random matrix realization of the ensemble $\Gamma(\mathrm{R}_i, \mathrm{L}_i, \mathrm{N})$ with size $\mathrm{M}_i \times \mathrm{N}$.

(2) *A sampling step*: a vector of M_i tests $\mathbf{y}_i$ is generated as $\mathbf{y}_i = \mathrm{q}_o(\mathbf{A}_i, \mathbf{x}_0)$ where $\mathrm{q}_o(\cdot,\cdot)$ is the OR-encoder introduced in equation 1.5.

(3) *A decoding step*: an attempt is made to infer the labels of the subjects of the population employing the erasure decoder described in algorithm 2.1 of chapter 2: $\mathbf{x}_i = \mathrm{g}_e(\mathrm{A}(i), \mathrm{y}(i))$[1], where $\mathrm{y}(i)$ and $\mathrm{A}(i)$ are the succession of tests and the succession of matrices generated up to the i-th stage

$$\mathrm{y}(i) \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_i]$$
$$\mathrm{A}(i) \triangleq [\mathbf{A}_1, \ldots, \mathbf{A}_i]$$

At the beginning of the encoding process all subjects of the population are tested. In subsequent stages the tests focus on those subjects that cannot be labeled by the tests in the previous stages.

1 Note that the erasure decoder is required to reconstruct the graph at the decoder side. Once the graph is reconstructed, the decoder can employ the decoding algorithm in appendix 2.A.2 for best performance, see figure (5.4(b)) in section 5.2 to asses the performance gain.

The process goes on until all subjects are labeled (perfect recovery) or it is decided that the fraction of subjects that remain unlabeled is small enough (lossy scheme).

1 **Initialize**: $i = 1$, $\mathcal{U}_0 = \{1,...,\mathrm{N}\}$,
2 **while** $|\mathcal{U}_{i-1}| > 0$ **do**
3 Design and construct $\mathbf{A}_i$
4 $\mathbf{y}_i = \mathrm{q}_o\big(\mathbf{A}_i, \mathbf{x}_0\big)$
5 $\big(\mathcal{U}_i, \mathcal{N}_i, \mathcal{D}_i\big) \leftarrow \mathrm{g}_e\big(\mathrm{A}(i), \mathrm{y}(i)\big)$
6 $i = i + 1$
7 **end**

Algorithm 5.1: Sequential multi-stage sampling

After a certain number of stages I a total number of tests $\sum_{i=1}^{I} \mathrm{M}_i$ has been done and the tests $\mathrm{y}(I)$ have been obtained as $\mathrm{y}(I) = \mathrm{q}_o\big(\mathrm{A}(I), \mathbf{x}_0\big)$.

At the end of the i-th encoding stage the population is partitioned into three disjoint sets:

Definition 5.1 (Set of subjects of the population that remain unlabeled at the end of the i-th stage, $\mathcal{U}_i$)**.**

$$\mathcal{U}_i \triangleq \big\{\forall j \in \mathcal{U}_i : \ x_j = '?'\big\}.$$

We define the complementary set as $\mathcal{U}_i^c \triangleq \mathcal{P} \setminus \mathcal{U}_i$.

Definition 5.2 (Set of defective subjects of the population that are labeled by the decoder at the end of the i-th stage, $\mathcal{D}_i$)**.**

$$\mathcal{D}_i \triangleq \mathcal{P}_D \cap \mathcal{U}_i^c,$$

Definition 5.3 (Set of non-defective of subjects the population that are labeled by the decoder at the end of the i-th stage, $\mathcal{N}_i$:)**.**

$$\mathcal{N}_i \triangleq \mathcal{P}_N \cap \mathcal{U}_i^c.$$

We also define their complementary sets as $\mathcal{D}_i^c \triangleq \mathcal{P}_D \setminus \mathcal{D}_i$ and $\mathcal{N}_i^c \triangleq \mathcal{P}_N \setminus \mathcal{N}_i$.

Next we focus on lines 3 and 5 of algorithm 5.1 because the testing stage is the same than in the non-adaptive group testing scheme, described in section 4.3.

5.1.2 Design and construction step

In the design and construction step of the i-th stage ensemble $\Gamma(\mathrm{R}_i,\mathrm{L}_i,\mathrm{N})$ is selected, i.e. the variable and check node degree profiles $\mathbf{A}_i$ (denoted by $\mathrm{R}_i(d)$ and $\mathrm{L}_i(d)$) are decided. Afterwards a new piece of the measurement matrix $\mathbf{A}$ is created: the sub-matrix $\mathbf{A}_i$ with dimensions $\mathrm{M}_i \times \mathrm{N}$, being M_i the number of tests done in the i-th stage.

This sub-matrix is created using the information from the previous stages according to the following rules:

Remark 5.1. *All subjects of the population $\mathcal{P}$ keep being tested until they are labeled.*

In other words, at the i-th stage the subjects indexed in $\mathcal{U}_{i-1}$ are the only ones that contribute to generate the tests $\mathbf{y}_i$. In order to do so,

Remark 5.2. *Each subject that remains unlabeled at the end of the $(i-1)$-th stage contributes to exactly one test in the following stage.*

On the one hand, remark 5.1 and 5.2 motivate the search space reduction, i.e. the reduction of the amount of components in $\mathcal{U}_i$ as i increases. Doing so the testing efforts are targeted only to the subjects that remain unlabeled after the $(i-1)$-th stage. This matrix $\mathbf{A}_i$ is chosen at random among those matrices in the ensemble $\Gamma\left(\mathrm{R}_i,\mathrm{L}_i,\mathrm{N}\right)$ that have all non-zero coefficients placed in the columns corresponding to unlabeled subjects $\mathcal{U}_i$.

The purpose of choosing $\mathbf{A}_i$ randomly from the ensemble $\Gamma(\mathrm{R}_i,\mathrm{L}_i,\mathrm{N})$ is twofold. First, matrix design is kept simple. Second, matrix performance can be predicted from the analysis presented in chapter 6.

It can be easily shown [129] that the i-th stage contributes to the overall rate with r_i:

$$r_i = \frac{\mathrm{M}_i}{\mathrm{N}} = \frac{\sum_{\forall d \geq 0} d\mathrm{L}_i\left(d\right)}{\sum_{\forall d \geq 0} d\mathrm{R}_i\left(d\right)} = \frac{\bar{d}_v^i}{\bar{d}_c^i} \tag{5.1}$$

where $\bar{d}_v^i$ and $\bar{d}_c^i$ are the average fraction of edges per variable and check node of the graph $\mathbf{A}_i$.

Variable node degree profile

The variable node degree profile distribution at the i-th stage $\mathrm{L}_i(d)$ is constrained by the scheme construction, since only the unlabeled subjects at the end of the previous stage $\mathcal{U}_{i-1}$ are further processed, i.e. all nodes associated to labeled subjects have degree i. So, we have only to decide the variable node degree profile of the nodes in $\mathcal{U}_{i-1}$. For simplicity we assumed that each

unlabeled subject participates in one test per stage, so the variable node degree distribution can be written in terms of the amount of elements of the set as

$$\mathrm{L}_i(d) = \begin{cases} 1 - \frac{|\mathcal{U}_{i-1}|}{\mathrm{N}} & d = 0 \\ \frac{|\mathcal{U}_{i-1}|}{\mathrm{N}} & d = 1 \end{cases} \tag{5.2}$$

which in the N-asymptotic regime becomes

$$\mathrm{L}_i(d) = \begin{cases} \mathrm{P}(\mathcal{U}_{i-1}^c) & d = 0 \\ \mathrm{P}(\mathcal{U}_{i-1}) & d = 1 \end{cases} \tag{5.3}$$

Check node degree profile: design of policy

To the best authors knowledge, there are no results in the literature on the design of matrix $\mathbf{A}_i$ for adaptive GT scheme, so design of a policy to choose $\mathrm{R}_i(d)$ is an open problem that still remains to be addressed.

For simplicity we choose to design a probability distribution $\mathrm{R}_i(d)$ based on results obtained by the decoder at the end of the stage $i-1$, i.e $\mathcal{U}_{i-1}$, $\mathcal{D}_{i-1}$ and $\mathcal{N}_{i-1}$.

Even doing this simplification there are infinite different designs for the degree profile of the different stages that yield the same aggregated rate $\sum_{i=1}^{I} \mathrm{M}_i/\mathrm{N}$, so there are infinite possible designs of the adaptive GT scheme.

In [119] we proposed a policy to design the degrees that will be shown in chapter 6 to have good performance. This policy aims the design of the matrix $\mathbf{A}_i$ to balance the number of ones and zeros of the tests vector $\mathbf{y}_i$. The rationale of this policy is that it maximized the entropy of $\mathbf{y}_i$, since the tests in $\mathbf{y}_i$ are independent because they operate on disjoint sets of subjects. Hence, if it were true that the tests $\mathrm{y}(i)$ were independent then $\mathrm{H}(\mathrm{y}(i))$ would be maximized and the proposed policy would provide the most informative results (it would maximize $\mathrm{I}(\mathbf{x}_0, \mathrm{y}(i))$. Unfortunately, the tests $\mathrm{y}(i)$ are not independent so this policy cannot be claimed to be optimal from the entropy point of view.

In order to have tests with probability 0.5 of being defective we must choose the degree profile to satisfy (5.6) which is the generalization of equation equation (2.4) for irregular degree profiles. In this equation $\mathrm{P}\big(\mathcal{N}_{i-1}^c \;\big|\; \mathcal{U}_{i-1}\big)$ is an estimate of the fraction of defective subjects that still remain indexed in $\mathcal{U}_{i-1}$. To further simplify the degree profile design, we decided to constrain the check degree to take only two consecutive values, i.e. $\mathrm{R}_i(d)$ is non-zero only for two values d' and $d'+1$. Although there is no optimality claim in this design, our experience has shown that other choices lead to a performance reduction due to an increase in Type III check nodes. Further details on this policy can be found in appendix 5.A.

This approach to design $\mathrm{R}_i(d)$ regards on the knowledge of $\mathrm{P}\big(\mathcal{D}_{i-1}^c \mid \mathcal{U}_{i-1}\big)$ at the end of each stage. If we know the amount of defective subjects in the population K we can simply obtain it as

$$\mathrm{P}\big(\mathcal{N}_{i-1}^c \mid \mathcal{U}_{i-1}\big) = 1 - \frac{\mathrm{K} - |\mathcal{D}_{i-1}|}{|\mathcal{U}_{i-1}|}. \tag{5.4}$$

otherwise we need an estimate of it obtained from the a priori information.

Matrix construction

After selecting the ensemble of matrices $\Gamma\big(\mathrm{R}_i,\mathrm{L}_i,\mathrm{N}\big)$ we can move to construct a matrix, first selecting a check node degree d according to distribution $\mathrm{R}(d)$ and then selecting d unlabeled subjects of the population $\mathcal{U}_{i-1}$ with uniform probability.

This matrix $\mathbf{A}_i$ is chosen among all matrices in the ensemble $\Gamma\big(\mathrm{R}_i,\mathrm{L}_i,\mathrm{N}\big)$ where all the non-zero coefficients are located in the columns corresponding to unlabeled subjects.

5.1.3 Decoder

As indicated previously, during the encoding process the ED is employed at the end of each stage. If the process continues until full recovery, no other decoder is necessary. However, the encoding process can be stopped at any stage i if a certain fraction of unlabeled subjects is tolerated. In that case, the fraction of unlabeled subjects can be reduced by applying a more sophisticated decoder like the BP-based decoder presented in section 2.2.2 to matrix $\mathrm{A}(i)$ and tests $\mathrm{y}(i)$. Note that in that case some subjects might be erroneously labeled.

In section 5.2 of simulation results we compare the performance of this adaptive sampling scheme when only the erasure decoder is employed to label subject with the performance of the same adaptive sampling scheme that implements a last attempt to retrieve all the labels of the subjects with the BP-based decoder (the LDPrC decoder adapted to the OR-operation) reviewed in section 2.2.2.

5.2 Performance of the proposed adaptive group testing scheme

Here we compare the performance of the proposed adaptive group testing scheme in two different setups. First, we compare it with the performance of a non-adaptive scheme for sparse pattern recovery in section 5.2.1. Second, within the source compression framework, we compare it with the rate distortion bound in section 5.2.2. Note that for the sake of clarity we use the group testing notation in the section 5.2.1 whereas we employ source coding notation in section 5.2.2.

The proposed adaptive scheme can be employed under two different premises. First, for perfect recovery, i.e. to label all subjects of the population. In this situation stages are added and more samples generated until the algorithm labels all subjects of the population. In the second case one establishes a fixed maximum number of samples, i.e. the maximum number of tests is prefixed, so the algorithm is stopped when the number of samples reaches the maximum allowed leading to in an incomplete recovery. We measure this incompleteness with the following metric

$$e = e(\mathbf{x}_0,\mathbf{x}) = \frac{|\mathcal{D}_i|}{\max\left(|\mathcal{P}_D|,|\mathcal{D}_i \cup \mathcal{U}_i|\right)} \tag{5.5}$$

where $|\mathcal{D}_i \cup \mathcal{U}_i|$ accounts for either the amount of subjects that remain unlabeled and the amount of defective subjects that are labeled. This metric varies when the defective subjects are labeled since the metric only changes when new defective subjects are labeled at the end of the stage. Note that e=1 indicates that all subjects were labeled.

We compare the sampling or testing efficiency of the proposed group testing scheme of the proposed scheme with $\mathrm{K}\log(\mathrm{N/K})$, since the algorithm for sparse pattern recovery and adaptive group testing required a number of tests of that order to achieve a perfect reconstruction (see section 2.1.2 for further details). Thus, for the sake of comparison we define the gap to this bound as

$$\text{gap} = \frac{\bar{\mathrm{M}}}{\mathrm{K}\log(\mathrm{N/K})}$$

where $\bar{\mathrm{M}}$ is the average number of tests required by the proposed adaptive scheme for perfect reconstruction.

5.2.1 Group testing performance

Figure (5.1) shows simulation results of the proposed adaptive scheme (P) for perfection recovery for populations of size $\mathrm{N} = 500$ and $\mathrm{N} = 10000$ and different number of defective subjects K where the degree profiles were designed according to the policy described in appendix 5.A. Each point of the plots was obtained by Monte Carlo with $2 \cdot 10^5$ simulations and K was assumed to be known.

In figure (5.1.a) we compare the evolution of the average number of tests $\bar{\mathrm{M}}$ required for perfect reconstruction of the proposed adaptive scheme (P) with evolution of the bound $\mathrm{K}\log(\mathrm{N/K})$ (B). We can observe that there exists an approximately constant multiplicative factor with respect to the bound which is larger than 1[1].

1 gap ≈ 1.5 for $\mathrm{N} = 500$ and gap ≈ 1.4 for $\mathrm{N} = 10000$.

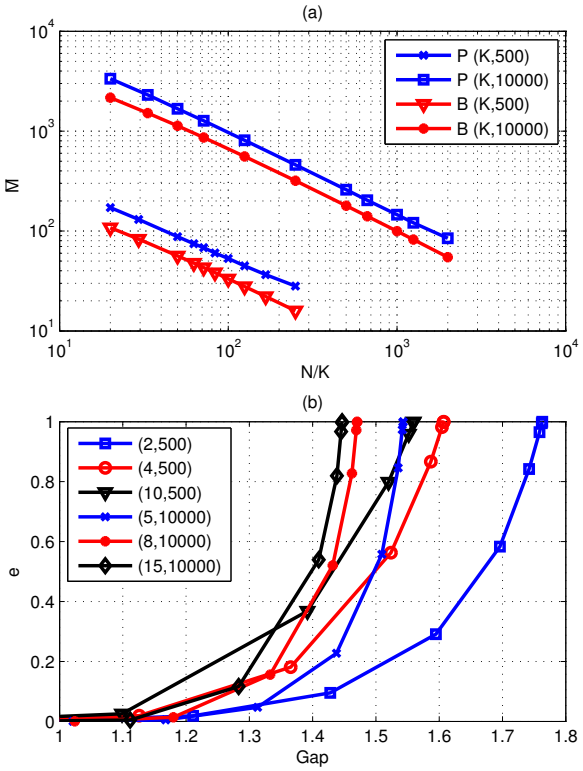


Figure 5.1: (a) Average number of tests $\bar{M}$ for perfect recovery, for fixed N and varying the sparsity K of the proposed scheme (P) compared with K log(N/K) (B). (b) Evolution of the error metric for a given population (K,N) as the number of stages increases as a function of the gap respect to K log(N/K).

Figure (5.1.b) shows the evolution of the recovery metric versus the gap through the stages of the adaptive group testing algorithm, i.e. for an increasing number of measurements. The points of the curves represent the resulting gap and recovery metric at the end of a stage. We have chosen to represent the gap instead of the number of samples because the normalization with the bound allows us to observe the dependence of the performance on (K,N): in a closer look, the gap decreases when the amount of defective subject K increases for a fixed N, and also decreases with N.

The plotted curves help us to explain the behavior of the sequential process: while the number of tests is below the bound (i.e. $g < 1$), only non-defective subjects of the population are labeled, so the recovery metric stays at 0. However, the knowledge of the non-defective produces an avalanche effect when the number of samples exceeds a threshold. Next stages rapidly determine the defective subjects in the population until recovery is perfect.

Figure (5.2.a) compares the performance of the proposed algorithm when K is exactly known

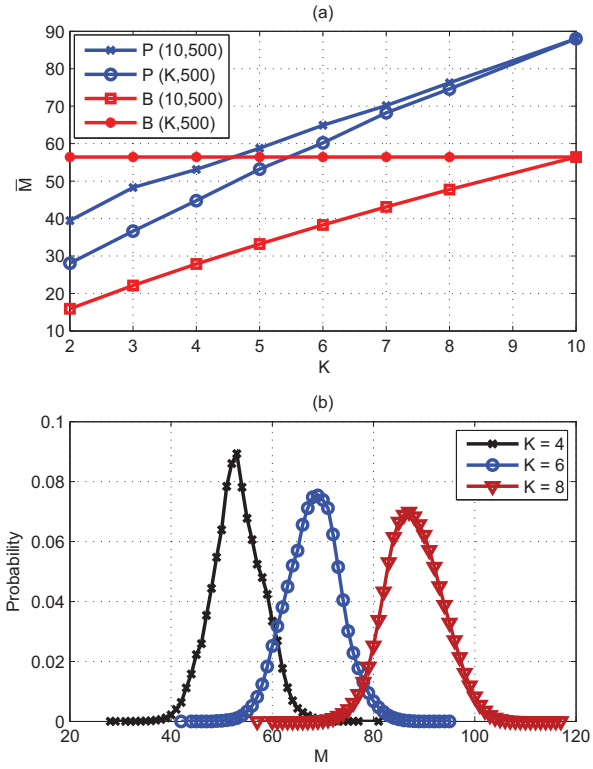


Figure 5.2: (a) Average number of samples $\bar{M}$ for perfect recovery for N = 500 comparing results when K is known (P(K,500)) with the ones when and upper value $K^* = 10$ is known (P(10,500)). The bound (B) K log(N/K) is included for comparison when K is both known or only an upper bound is known. (b) Probability Mass function of the number of samples for N = 500 and several K.

with the case in which only an upper bound K^* is known. The figure depicts the comparison of the average number of tests required by the proposed scheme (P) for perfect recovery with the bound K log(N/K) (B) when N = 500 and K is either known at the decoder side (K,500) or only an upper bound of K is known (10,500). The figure shows that the performance of the proposed scheme falls when the exact number of defective subjects is unknown and assume an upper bound $K^* = 10$, at the expense of adding a limited number of additional tests (the more the further is K from K^*).

Figure (5.2.b) shows the probability mass function of the number of samples for a given pair (K,N). Two million samples from each distribution were simulated for K = {4,6,8} and N = 500. It can be seen that the variance increases with the K and also the probability mass function becomes more Gaussian-shaped. Notice that the probability mass is concentrated around the mean; this means that the event of requiring a much larger number of measurements to recover the unlabeled subjects is very unlikely. In the simulations, none of the realizations required more

than 81, 95 and 117 samples for K = 4,6,8, respectively.

We can also compare the results in figure (5.2.a) with the simulated performance of several non-adaptive group testing schemes in ([5], figure 3 and 4). Their best performing scheme employs a Bernoulli matrix with $p = 1/10$ and employs the decoding algorithm named smallest satisfying set (SSS). For (K = 10,N = 500) this algorithm requires at least 100 tests to reconstruct perfectly a 50 % of the realizations of the population whereas our proposed adaptive algorithm requires only 88 tests[1].

Table 5.1: Comparison with OMP [151] in terms of required number of tests for perfect reconstruction when N = 256, for several K.

K	M_{OMP}	M^*	$\bar{M}$
4	56	50	39.3
8	96	76	64.0
12	136	98	85.2
16	184	117	104.4
20	228	135	121.4

We compare with the simulations in [151][2] obtained with the orthogonal matching pursuit (OMP) reconstruction algorithm with fixed M = 256 and several K with the proposed scheme when K is known. Table 5.1 gathers the number of tests required for OMP (M_{OMP}) with the average and the maximum number of samples ($\bar{M}$ and M^*) required by the proposed method. A significant improvement can be observed.

In table 5.2 we compare the average number of tests required for perfect reconstruction (we assume that reconstruction is perfect when $P(\mathcal{U}_i) < 10^{-6}$) with the number of samples required for perfect reconstruction in the worst case scenario of the adaptive group testing, based on random walk [35, 77], for different pairs of (N,K) and assuming the K is known.

Table 5.2: Performance comparison of the proposed adaptive group testing scheme ([P]) and the random walk- based algorithms in [35, 77]

N	10^5	10^5	10^5	10^5	10^5
K	996	5666	12666	18	180
M [P]	9338	36730	63950	154	1540
M [77]	8812	37072	67478	144	1475
M [35]	8398	34798	63089	140	1403

Note that the comparison between the proposed algorithm and the other algorithms is not fair

1 Note that the average number of tests coincides with the 50% probability of perfect reconstruction.
2 Note that [151] presents the results in the framework of the sparse pattern recovery.

since the algorithms presented in [35, 77] mark as non-defective all the subjects remaining to be labeled once all the defective subjects have been labeled, since K is known, whereas the proposed algorithm keeps at generating tests while there are still unlabeled subjects. Furthermore, note that the performance of the proposed scheme will change if the policy to choose the ensemble changes. So, the results in table 5.2 are merely indicative and show that the adaptive scheme has similar performance that the recently proposed adaptive algorithm in [35] outperforms to [77] in the worst case scenario, and that the performance of the algorithm in [77] and our proposed algorithm apparently are close.

5.2.2 Binary source compression performance

Now let us analyze the performance of the application of the adaptive scheme proposed in this chapter within the binary source compression framework, where the compression limit is the rate distortion bound[1] 2.1.

Here we slightly modified the proposed scheme: the matrices are still designed at the beginning of each stage but tests are generated one by one (we do not skip any tests) instead of generating the i-th block of test at once. Therefore, after acquiring each test the decoder is activated and attempts to label as many subjects as possible. Doing so we obtain smoother performance curves.

We show the performance of the proposed scheme in terms of bit error rate (BER) and word error rate (WER), which can be regarded as the average fraction of unlabeled subjects and the average fraction of population realizations (simulations of the experiment) that have been completely labeled for a given rate, respectively.

Figure 5.3 depicts the performance of the algorithm for different sources and N. Each plot was obtained by averaging the results of at least 100000 simulations. The plots represent the average BER obtained for a given rate indicated in the horizontal axis and were obtained simply averaging the amount of error bits for a given number of samples and dividing by N.

Figure 5.3 shows that the rate required to achieve the free error zone is slightly larger for shorter N; this is the penalty for using small N. Note that in the case of $\mathrm{P}(X = 1) = 0.012$ and $\mathrm{K} = 1000$, the average number of ones in $\mathbf{x}_0$ is only 12. This figure also shows that the system requires an overhead respect to the entropy to achieve free error zone, e.g. for $\mathrm{P}(X = 1) = 0.018$ and $\mathrm{N} = 1000$ and 10000, the average rates are 0.1120 and 0.1153 whereas the entropy rate is 0.0938. These results can be compared with the ones in section 4.4.4, where a similar scheme with fixed

1 We know that the comparison is unfair since the rate-distortion bound cannot be achieved by the group testing scheme but we decided to enclose it for the sake of clarity.

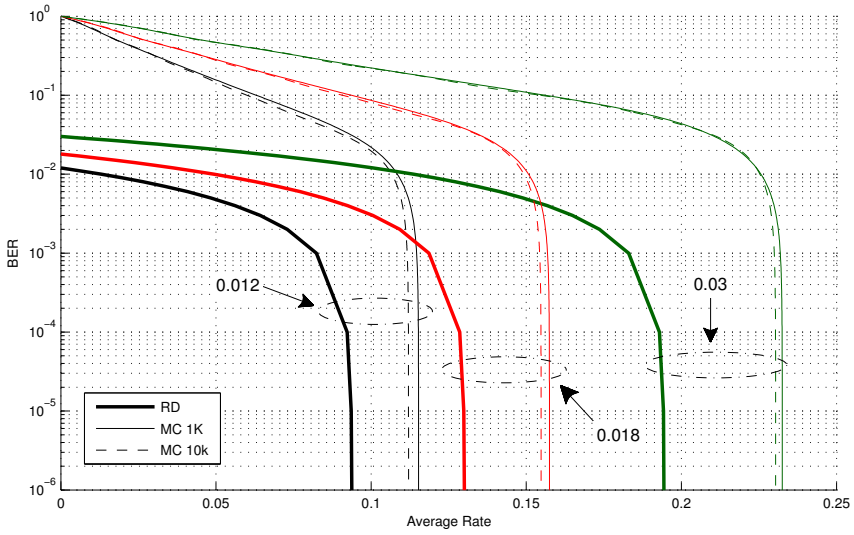
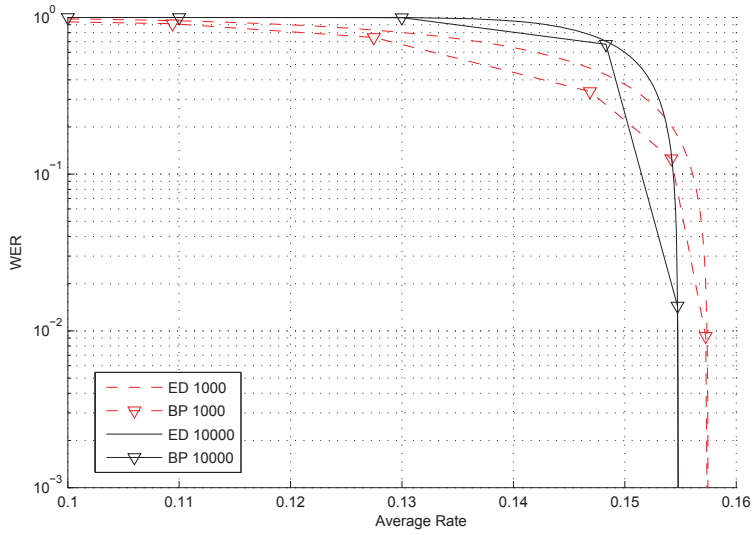


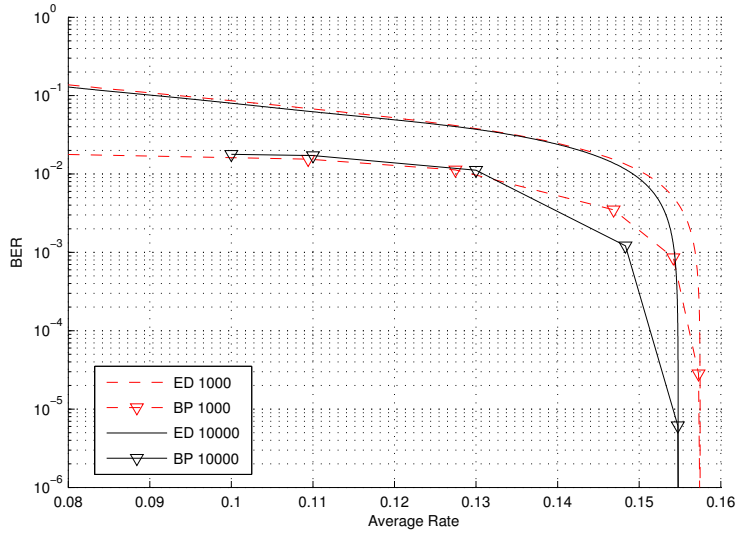
Figure 5.3: Comparison of the performance of the adaptive algorithm obtained by Monte Carlo simulation (MC) for N = 1000 (1k) and N = 10000 (10K) bits with the rate distortion bound (RD) for different probabilities $P(X = 1) = 0.012,\ 0.018,\ 0.03$

block codes at each stage required and a rate of 0.1191 to achieve a BER of 10^6 for a N = 100000 with $P(X = 1) = 0.013$ (see tables 4.1 and 4.2).

Figure 5.4(a) and 5.4(b) show the performance gain introduced when the erasure decoder is replaced by the optimum BP decoder for signal reconstruction (LDPrC-based decoder, reviewed in section 2.2.2). Figure 5.4(b) shows that the BP decoder provides lower distortion when the compression rate approaches the limit, but both decoders have the same performance when a low distortion is desired. It must be noted that these schemes do not exhibit error floor.



(a) Comparison of BP and ED. Word error rate (WER)



(b) Comparison of BP and ED. Bit error rate (BER)

Figure 5.4: Comparison of performance of adaptive scheme with erasure decoder (ED) or belief propagation-based (BP) decoder for $\mathrm{P}(X = 1) = 0.018$ and $\mathrm{N} = 1000,\ 10000$.

Appendix 5.A Check degree profile design

The policy is aimed to maximize the entropy of the binary sequence $\mathbf{y}_i$. So a first constraint arises from equation (2.4) since the average fraction of Type I check nodes must be equal to 0.5.

Then it follows that ensemble must have a check node degree distribution $\mathrm{R}_i(d)$ so as

$$\sum_{\forall d} \mathrm{R}_i(d)\, \mathrm{P}\left(\mathcal{N}_{i-1}^c \mid \mathcal{U}_{i-1}\right)^d = 0.5 \tag{5.6}$$

where the term $\mathrm{P}\left(\mathcal{N}_{i-1}^c \mid \mathcal{U}_{i-1}\right)$ is known at instant i in case that K. For the sake of simplicity in this appendix we refer to this term simply as the probability p.

We decided to add an extra restriction by choosing ensembles with at most check nodes with two different number of edges, i.e. $\mathrm{R}_i(d)$ have at most two non-null coefficients. Then equation (5.6) becomes

$$\mathrm{R}_i\left(d'\right) p^{d'} + \left(1 - \mathrm{R}_i\left(d'\right)\right) p^{d''} = 0.5 \tag{5.7}$$

where $d' < d''$ are employed degrees, d', $d'' \in \mathcal{N}$, and $\mathrm{R}_i(d')$ is the fraction of check nodes with degree d'. Note that the equation has solutions and hence there are infinite many ensembles that accomplish these two restrictions.

For simplicity we decided to add the extra constrain $d'' = d' + 1$. Then if we apply it to equation (5.7) and operate it results

$$d' = \frac{\log 0.5}{\log p} - \frac{\log\left(p + \mathrm{R}\left(d'\right)(1-p)\right)}{\log p} \tag{5.8}$$

This equation has a single solution because when $d' \in \mathbb{N}$ and $\mathrm{R}(d') \in [0,1]$ it follows that d' is the natural number sandwiched by

$$\frac{log\,(0.5)}{\log\left(\mathrm{P}\left(\mathcal{N}_{i-1}^c \mid \mathcal{U}_{i-1}\right)\right)} \geq d' \geq \frac{log\,(0.5)}{\log\left(\mathrm{P}\left(\mathcal{N}_{i-1}^c \mid \mathcal{U}_{i-1}\right)\right)} - 1 \tag{5.9}$$

and then, the term $p + \mathrm{R}(d')(1-p) \geq p$ implies that the term $0 < \frac{\log\left(p+\mathrm{R}(d')(1-p)\right)}{\log p} < 1$. So, the term $\mathrm{R}_i(d)$ can be easily foundapplying the result in (5.9) to equation (5.7) and rearranging terms

$$\mathrm{R}_i\left(d'\right) = \frac{\frac{0.5}{p^{d'}} - p}{1-p} \tag{5.10}$$

and so the term $\mathrm{R}(d'+1) = 1 - \mathrm{R}_i(d')$.

CHAPTER 6

Adaptive group testing and decision processes

The design of adaptive group testing schemes is challenging because of the intricate relationship between the sampling matrix employed in each stage and the overall system performance. For example, it is hard to say whether it is better to include high-degree test nodes in early stages or it is better to use low-degree tests. In the first case, those nodes will have little impact on the overall rate but probably positive tests will not be helpful to label any defective subjects in later stages (in terms of the notation of section 2.3.3, the fraction of Type-III nodes in equation 2.6 will probably be high). In the second case, the impact on efficiency will be greater but the tests are more likely to allow labeling subjects (the fraction of Type-I and Type-II nodes in equation 2.4 and 2.4 will be higher), so they might be more useful in the long run. Hence, it would be very convenient to envisage a design tool that allowed evaluating the performance of an adaptive GT scheme without need of actually implementing it to run Monte Carlo simulations. If this tool were available, different GT designs could be easily compared.

The multiple stages of adaptive GT system can be designed employing policies with different degrees of complexity. Our analysis is targeted at assisting the design of adaptive GT schemes where the testing matrices are randomly generated as belonging to a certain matrix ensemble, as was described in chapter 5. A more sophisticated policy would be to design each new stage as a structured matrix selected taking into account the degree profile of the aggregated testing matrix including all previous stages and taking into account the sets of defective/non-defective subjects labeled at each stage (e.g. it could take into account which variable nodes are more likely to be defective based on past history and link them to lower degree checks). Although the latter policy could potentially lead to better performance, we have not taken it into consideration in our analysis. As shown in this chapter, by constraining ourselves to work with randomly generated matrices we are capable of predicting very accurately the performance of the resulting adaptive GT system using probabilistic tools.

In this chapter we propose a systematical framework to characterize the evolution of the adaptive group testing scheme presented in chapter 5 that poses adaptive GT as a non-Markovian decision process. By casting adaptive GT into the decision problem framework, we provide an analytic tool that predicts the performance of an adaptive GT system employing as inputs the ratio of defective subjects $\mathrm{P}(\mathcal{P}_D)$ and the degree profiles of the matrix ensembles employed in each stage $\Gamma\left(\mathrm{L}_i,\mathrm{R}_i,\mathrm{N}\right)$.

The advantage of the proposed analytic tool is twofold. First, it can be employed in a real system implementation to decide how to design the next testing stage based on the outcome of the previous ones. Second, it can also be employed to predict the performance of the adaptive GT scheme if a specific combination of matrix degree profiles $\left\{\Gamma\left(\mathrm{L}_1,\mathrm{R}_1,\mathrm{N}\right),\Gamma\left(\mathrm{L}_2,\mathrm{R}_2,\mathrm{N}\right),\ldots\right\}$, etc. is employed in successive stages.

This chapter is divided as follows: in section 6.1 we formulate the adaptive group testing system described in chapter 5 as a decision problem. In section 6.2 we derive the equations employed to predict the system performance in terms of the fraction of defective subjects that remain unlabeled after a certain number of testing stages. The analysis is done for the N-asymptotic regime. Nevertheless, the simulation results included in section 6.3 evidence that the analysis is also useful to predict system performance for relatively small population sizes.

6.1 Decision processes and adaptivity

6.1.1 Introduction

Decision processes are widely employed to model decision making situations where the outcomes are partly random and partly under the control of the agent that makes the decision [86, 113]. In the case of interest, each stage in the adaptive GT scheme can be regarded as a step in the decision process. Then, the state at a certain stage i, s_i, is somehow related to the amount of defective subjects that remain unlabeled after i testing stages have been performed. Besides, the transition between the state at instant $i-1$ and at i occurs when the decision-maker (agent) chooses the testing matrix employed in the i-th stage (the action), i.e. either it chooses $\mathbf{A}_i$ or the ensemble to which the matrix belongs $\Gamma\left(\mathrm{L}_i,\mathrm{R}_i,\mathrm{N}\right)$. When the action is taken, test results $\mathbf{y}_i$ are obtained (the observation). For simplicity we define the successions of states, actions and

observations as $\mathrm{s}(i)$, $\mathrm{A}(i)$ or $\Gamma(i)$, and $\mathrm{y}(i)$[1], respectively, as follows

$$\begin{aligned}
\mathrm{s}(i) &\triangleq \{s_1, s_2, \ldots, s_{i-1}, s_i\} \\
\mathrm{A}(i) &\triangleq \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{i-1}, \mathbf{A}_i\} \\
\Gamma(i) &\triangleq \{\Gamma(\mathrm{L}_1, \mathrm{R}_1, \mathrm{N}), \Gamma(\mathrm{L}_2, \mathrm{R}_2, \mathrm{N}) \ldots, \Gamma(\mathrm{L}_{i-1}, \mathrm{R}_{i-1}, \mathrm{N}), \Gamma(\mathrm{L}_i, \mathrm{R}_i, \mathrm{N})\} \\
\mathrm{y}(i) &\triangleq \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{i-1}, \mathbf{y}_i\}
\end{aligned}$$

Hence, adaptive GT design is formulated as the problem of finding a suitable decision process model that can assist the agent in the selection of the testing matrices to be employed at each stage. This decision process requires also the characterization of the state transitions produced by the actions. The non-Markovian nature of this decision process stems from decoding process described in 2.3.2: the labeling of defective subjects in the i-th instant depends on the actions and test results from previous instants, as stated by remark 2.2.

6.1.2 State definition

Different state definitions can be employed that assume different degrees of knowledge of the population subsets:

(1) $s_i \triangleq \{\mathcal{P}_D, \mathcal{U}_i\}$

This state definition requires a genie agent that has access to full knowledge of the sets $\mathcal{P}_D$ (and $\mathcal{P}_N$) and of the subjects that remain to be labeled after i rounds $\mathcal{U}_i$. While this definition has no real use (if the agent knew $\mathcal{P}_D$ before testing started then no measurements would be required), it is useful for illustration purposes.

This state definition is the only one for which the decision-making agent can have full certainty about what the ending state at instant i will be if a certain action $\mathbf{A}_i$ is taken. In other words, the following probabilities are indicator functions (they are either 0 or 1) and are identical:

$$\begin{aligned}
\mathrm{P}\Big(S_i = s_i \;\Big|\; \mathrm{s}(i-1),\ \mathrm{A}(i),\ \mathrm{y}(i),\ s_0\Big) &= \mathrm{P}\Big(S_i = s_i \;\Big|\; \mathrm{s}(i-1),\ \mathrm{A}(i),\ s_0\Big) \\
&= \mathrm{P}\Big(S = s_i \;\Big|\; \mathrm{A}(i),\ \mathrm{y}(i),\ s_0\Big)
\end{aligned}$$

Note that, as it is a non-Markovian process, $\mathrm{P}\Big(S_i \;\Big|\; s_{i-1},\ \mathbf{A}_i,\ \mathbf{y}_i\Big)$ is not an indicator function.

1 Note that $\mathrm{y}(i)$ and $\mathrm{A}(i)$ were already defined in chapter 5 as the succession of tests and matrices. Within this new context we refer to them as succession of observations and actions.

(2) $s_i \triangleq \left\{|\mathcal{D}_i^c|,\mathcal{U}_i\right\}$

A more realistic state definition is to assume that the agent managed to know somehow the amount of defective subjects in the population, $|\mathcal{P}_D|$. Hence, after finishing the i-th stage, the agent would know the amount of defective subjects that remain unlabeled, but it would not know their location (it would only know they are in $\mathcal{U}_i$). Due to this uncertainty, the probability distribution $\mathrm{P}\Big(S_i \;\Big|\; \mathrm{s}(i-1),\ \mathrm{A}(i),\ s_0\Big)$ is not an indicator function, since the same action can lead to transitioning into different states depending on the location of the unlabeled defective subjects. Nevertheless, the agent can know the state s_i after the action has been taken and the corresponding observation is available: the probability distributions

$$\mathrm{P}\Big(S_i \;\Big|\; \mathrm{s}(i-1),\ \mathrm{A}(i),\ \mathrm{y}(i), s_0\Big) = \mathrm{P}\Big(S_i \;\Big|\; \mathrm{A}(i),\ \mathrm{y}(i),\ s_0\Big)$$

are indicator functions.

This state definition provides a rather exhaustive description of the GT system that would be useful to track the system evolution when the observations $\mathrm{y}(i)$ are available. It would also be useful to design sophisticated adaptive GT schemes employing structured matrices like those that are mentioned in the introduction of the current chapter and are not considered in our analysis. However, one of the key features of the adaptive GT scheme proposed in this thesis is that matrices $\mathbf{A}_i$ are randomly picked from the ensemble $\Gamma\left(\mathrm{L}_i,\mathrm{R}_i,\mathrm{N}\right)$. Due to this design constraint, as the new check nodes added to the graph are linked randomly to the unlabeled variable nodes, the selection of the degree profile of the i-th stage does not depend on $\{\mathcal{U}_1,\ \ldots,\ \mathcal{U}_{i-1}\}$ but only on $\{|\mathcal{U}_1|,\ \ldots,\ |\mathcal{U}_{i-1}|\}$. Hence, from the point of view of designing the adaptive GT scheme, the state definition $s_i \triangleq \left\{|\mathcal{D}_i^c|,\mathcal{U}_i\right\}$ is overly complex. For design purposes, the next state definition does not entail any information loss.

(3) $s_i \triangleq \left\{|\mathcal{D}_i^c|,|\mathcal{U}_i|\right\}$

This state definition includes all the information needed to predict the performance of the adaptive GT scheme described in chapter 5 in the N-asymptotic regime. However, this state definition is not convenient for a probabilistic analysis because *i)* $|\mathcal{D}_i^c|$ and $|\mathcal{U}_i|$ can take infinitely large values in the N-asymptotic regime and their dynamic range changes with N and *ii)* $|\mathcal{D}_i^c|$ and $|\mathcal{U}_i|$ cannot be assumed independent since $|\mathcal{D}_i^c| \leq |\mathcal{U}_i|$, i.e. $\mathrm{P}(S_i = s)$ cannot be predicted by analyzing separately $\mathrm{P}(|\mathcal{U}_i|)$ and $\mathrm{P}(|\mathcal{D}_i^c|)$. Fortunately the next definition solves these problems.

Before we move on to the new definition, it is worth mentioning that the probability distribution $\mathrm{P}\big(S_i \;\big|\; \mathrm{A}(i),\ \mathrm{y}(i),\ s_0\big)$ is still an indicator function when the initial state s_0 is

known. Indeed, if $\mathrm{A}(i)$ and $\mathrm{y}(i)$ are known the encoding/decoding process can be reproduced, obtaining $\mathcal{U}_i$ (so its cardinality is known) and obtaining $\mathcal{D}_i$ (so $|\mathcal{D}_i^c| = |\mathcal{D}_o^c| - |\mathcal{D}_i|$). In this case $\mathrm{P}\big(|\mathcal{U}_i| \;\big|\; \mathrm{A}(i),\ \mathrm{y}(i),\ s_0\big)$ would also be an indicator function, obviously.

(4) $s_i \triangleq \left\{ \frac{|\mathcal{D}_i^c|}{|\mathcal{U}_i|}, \frac{|\mathcal{U}_i|}{|\mathcal{U}_0|} \right\}$

For practical purposes this definition is equivalent to the previous one, since there is a one-to-one mapping between any pair $\left\{|\mathcal{D}_i^c|,|\mathcal{U}_i|\right\}$ and $\left\{ \frac{|\mathcal{D}_i^c|}{|\mathcal{U}_i|}, \frac{|\mathcal{U}_i|}{|\mathcal{U}_0|} \right\}$ as long as $|\mathcal{U}_i|$ is not zero (remember that $|\mathcal{U}_0| = |\mathcal{P}| = \mathrm{N}$ is known). As the analysis focuses in the asymptotic case of infinite length, $|\mathcal{U}_i|$ might be small but will not be zero.

The pair of magnitudes included in the state definition represent now the fraction of defective subjects among those that remain unlabeled (i.e. the probability of an unlabeled subject being defective, $\mathrm{P}(\mathcal{D}_i^c \mid \mathcal{U}_i)$) and the fraction of the population that has not been labeled yet (i.e. the probability of a subject being unlabeled, $\mathrm{P}(\mathcal{U}_i)$) at the end of the i-th stage.

This definition solves the problems of the previous ones: *i*) now both components take values in the interval (0,1) and *ii*) $\mathrm{P}(|\mathcal{D}_i^c|, |\mathcal{U}_i|) = \mathrm{P}(|\mathcal{D}_i^c| \mid |\mathcal{U}_i|)\,\mathrm{P}(|\mathcal{U}_i|)$, so we can estimate the distribution of the two state components separately and combine them to get a state characterization without need of any approximation.

This is the state definition employed through the chapter.

The remaining of the chapter is devoted to the estimation of the adaptive GT process state after performing a certain amount of stages.

6.1.3 State estimation

Let us consider the implementation of the adaptive GT scheme described in chapter 5. Let us assume that the fraction of defective subjects in the population is known, so the initial state is known: $s_0 = \left\{ \mathrm{P}(\mathcal{P}_D),1 \right\}$. After i testing stages the agent designing the next testing stages has access to the testing matrices and to the test results obtained in the previous stages (i.e. it knows $\Gamma(i)$, $\mathrm{A}(i)$ and $\mathrm{y}(i)$). In order to select the matrix degree profile for the next j stages the agent takes into account the impact of the selection of ensembles $\left\{ \Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1},\mathrm{N}\right),\ \ldots,\ \Gamma\left(\mathrm{L}_{i+j},\mathrm{R}_{i+j},\mathrm{N}\right) \right\}$ on the probability distribution of the state at which the system will evolve after these actions

are taken: the agent must do a j-step prediction to estimate[1]:

$$\begin{aligned}\mathrm{P}\big(S_{i+j} = s_{i+j} \,\big|\, \Gamma\left(\mathrm{L}_{i+j},\mathrm{R}_{i+j},\mathrm{N}\right),\, \ldots,\, \Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1}\mathrm{N}\right),\, \mathrm{y}(i),\, \mathrm{A}(i),\, s_0\big) = \\ \mathrm{P}\big(\mathcal{D}^c_{i+j} \,\big|\, \mathcal{U}_{i+j},\, \Gamma\left(\mathrm{L}_{i+j},\mathrm{R}_{i+j},\mathrm{N}\right),\, \ldots,\, \Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1},\mathrm{N}\right),\, \mathrm{y}(i),\, \mathrm{A}(i),\, s_0\big)\times \\ \mathrm{P}\big(\mathcal{U}_{i+j} \,\big|\, \Gamma\left(\mathrm{L}_{i+j},\mathrm{R}_{i+j},\mathrm{N}\right),\, \ldots,\, \Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1},\mathrm{N}\right),\, \mathrm{y}(i),\, \mathrm{A}(i),\, s_0\big)\end{aligned}$$

For each choice of the ensembles to be employed in the following stages the state has a different probability distribution. In the N-asymptotic case this distribution converges to its mean, but for finite values of N it has a non-zero variance.

There are two particular instances of this probability distribution that correspond to interesting scenarios:

Scenario 1: Adaptive group testing implementation: $j = 1$

> After i testing stages, in order to design the next adaptive GT stage the agent analyzes the impact of the selection of $\Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1},\mathrm{N}\right)$ on the system evolution. In this case the magnitudes that must be estimated are $\mathrm{P}\big(\mathcal{D}^c_{i+1}|\mathcal{U}_{i+1},\, \Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1},\mathrm{N}\right),\, \mathrm{y}(i),\, \mathrm{A}(i),\, s_0\big)$ and $\mathrm{P}\big(\mathcal{U}_{i+1} \,\big|\, \Gamma\left(\mathrm{L}_{i+1},\mathrm{R}_{i+1},\mathrm{N}\right),\, \mathrm{y}(i),\, \mathrm{A}(i),\, s_0\big)$.

Scenario 2. Study and design of adaptive group testing policies: $i = 0$

> The agent analyzes the system evolution many stages ahead without actually implementing it. This is the task done to analyze how the ensemble design policy affects the performance of the adaptive GT scheme in the long run and to compare different policies and assess their impact on the evolution of the number of defective subjects that are labeled and the number of stages and number of tests that are required. In order to do this analysis, only the fraction of defective subjects in the population and the ensembles to be employed at each stage need to be known, since in this case the magnitudes that must be estimated are $\mathrm{P}\big(\mathcal{D}^c_j \,\big|\, \mathcal{U}_j,\, \Gamma(j),\, s_0\big)$ and $\mathrm{P}\big(\mathcal{U}_j \,\big|\, \Gamma(j),\, s_0\big)$.

Next, section 6.2 is devoted to the analysis of the probability distributions in these two scenarios in the N-asymptotic case or N sufficiently large. The extension to the most generic scenario is straightforward.

1 We omit to condition to the knowledge of $\Gamma(i)$ because it is redundant with conditioning to $\mathrm{A}(i)$.

First in subsection 6.2.2 Scenario 2 is considered obtaining the value of the state at which the system ends in the N-asymptotic regime[1]. Afterwards, in sub-section 6.2.3 it is shown how the equations for Scenario 1 are obtained by replacing some probability terms in the equations for Scenario 2 by the actual values measured in the adaptive group testing system implementation.

6.2 Predicting the evolution of the state distribution

The aim of this section is to show how to track the evolution of the $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i, \Gamma(i), s_0\big)$ and $\mathrm{P}\big(\mathcal{U}_i \mid \Gamma(i), s_0\big)$ as well as $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i, \Gamma_i, \mathrm{A}(i-1)\,\mathrm{y}(i-1), s_0\big)$ and $\mathrm{P}\big(\mathcal{U}_i \mid \Gamma_i, \mathrm{A}(i-1), \mathrm{y}(i-1), s_0\big)$. For the sake of conciseness we refer to these terms as $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$ and $\mathrm{P}\big(\mathcal{U}_i\big)$, respectively.

The amount of defective and non-defective subjects in the population ($|\mathcal{P}_D|$,$|\mathcal{P}_N|$) is fixed by the initial state s_0 and it is assumed to be known. Hence, initially both $\mathrm{P}(\mathcal{P}_D)$ and $\mathrm{P}(\mathcal{P}_N)$ are also known.

In sub-section 6.2.1 we define several sets and show the existing relations between them. These sets are used to characterize the evolution of the adaptive process in the N-asymptotic regime from instant i-1 to instant i.

6.2.1 Set definition

Here we define all the sets that we need to model the evolution of the parameters $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$ and $\mathrm{P}\big(\mathcal{U}_i\big)$ and we also show the relations between them that are relevant for the analysis.

As the agent applies the ED to all previous stages when designing the next one, to avoid confusion we denote as instant i the time when the i-th stage is designed and stage m, $1 \leq m \leq i$, refers to both the piece of matrix $\mathbf{A}_m$ on the vector of tests $\mathbf{y}_m$ at which the ED is applied at that instant. We start by defining

Definition 6.1. $\mathcal{D}_m^i$*: Set of defective subjects labeled at instant i from the sub-matrix* $\mathbf{A}_m$ *and the observations* $\mathbf{y}_m$ *when the non-defective subjects indexed in* $\mathcal{N}_i \setminus \mathcal{N}_m$ *are taken into account, as stated by remark 2.2. Note that* $\mathcal{D}_m^i = \emptyset$ *for* $i < m$ *and that* $\mathcal{D}_m^i$ *is equal to* $\mathcal{D}_{min}$ *of the m-th stage once all the non-defective subjects in the population are labeled.*

Definition 6.2. $\mathcal{N}_m^i$*: Set of non-defective subjects labeled at instant i from the sub-matrix* $\mathbf{A}_m$ *and the observations* $\mathbf{y}_m$*. Note that* $\mathcal{N}_m^i = \emptyset$ *for* $i < m$*.*

Definition 6.3. $\mathcal{N}_y^i$*: Set that indexes the non-defective tests of vector* $\mathbf{y}_i$*.*

1 Simulation results show that this value coincides with the mean of the state probability distribution in the N-finite case.

Note that $\mathcal{D}_m^i \subseteq \mathcal{U}_j$ and $\mathcal{N}_m^i \subseteq \mathcal{U}_j$ for $j < m$ and that they are only partially contained in any $\mathcal{U}_j$ for $j \geq m$ due to remark 5.2 and 5.1. Their expected (and desirable) behavior is that the amount of subjects indexed in the $\mathcal{D}_m^i$ and $\mathcal{N}_m^i$ increases as more and more actions are executed.

Definition 6.4. $\Delta\mathcal{N}_m^i$*: Set of non-defective subjects indexed in* $\mathcal{N}_m^i$ *that were not indexed at instant* $i-1$ *by the sets* $\mathcal{N}_m^{i-1}$

$$\Delta\mathcal{N}_m^i \triangleq \mathcal{N}_m^i \backslash \mathcal{N}_m^{i-1}$$

Due to remarks 5.1 and 5.2 $\Delta\mathcal{N}_m^m \equiv \mathcal{N}_m^m$ and $\Delta\mathcal{N}_m^i \equiv \emptyset \ \forall m < i$, since both the non-defective subjects are labeled as indicated by remark 2.1 and subjects are no further tested once they are labeled, as indicated by remark 5.1.

Definition 6.5. $\Delta\mathcal{D}_m^i$*: Set of defective subjects indexed in* $\mathcal{D}_m^i$ *that were not indexed at instant* $i-1$ *by the set* $\mathcal{D}_m^{i-1}$

$$\Delta\mathcal{D}_m^i \triangleq \mathcal{D}_m^i \backslash \mathcal{D}_m^{i-1}$$

The equivalence between $\Delta\mathcal{D}_m^i$ and $\mathcal{D}_m^i$ is not as straightforward as in the former case. For $i = m$, $\Delta\mathcal{D}_m^m \equiv \mathcal{D}_m^m$. Regretfully, as the instant i increases the set $\mathcal{D}_m^i$ is partitioned into two different subsets: the set $\Delta\mathcal{D}_m^i \cap \mathcal{U}_{i-1}$, $\Delta\mathcal{D}_m^i \cap \mathcal{D}_{i-1}$, because defective subjects can also be labeled using the tests from other stages. We are interested in tracking the subjects that are labeled in instant i by the defective check nodes of $\mathbf{A}_m$ but were not labeled in instants i-1, i.e. $\Delta\mathcal{D}_m^i \cap \mathcal{U}_{i-1}$.

Definition 6.6. $\Delta\mathcal{N}_i$*: Set of non-defective subjects that are labeled at instant* i *but were unlabeled at instant* $i-1$

$$\Delta\mathcal{N}_i \triangleq \bigcup_{1 \leq m \leq i} \Delta\mathcal{N}_m^i$$

Due to remarks 2.1 and 5.2 it holds that $\Delta\mathcal{N}_m^i \cap \Delta\mathcal{N}_k^i \equiv \emptyset \ \forall m \neq k$. Thus $\Delta\mathcal{N}_i$ is the union of several disjoint sets.

Definition 6.7. $\Delta\mathcal{D}_i$*: Set of defective subjects that are labeled at instant* i *but were unlabeled at instant* $i-1$

$$\Delta\mathcal{D}_i \triangleq \bigcup_{1 \leq m \leq i} \left\{\Delta\mathcal{D}_m^i \cap \mathcal{U}_{i-1}\right\}$$

Note that the intersection set $\Delta\mathcal{D}_m^i \cap \Delta\mathcal{D}_j^i \cap \mathcal{U}_{i-1}$, j,$m \leq i, m \neq j$, can be any non-empty set.

Due to the random construction of the measurement matrix $\mathrm{A}(i)$ all subjects indexed in $\mathcal{U}_{i-1}$ have the same probability of belonging to $\Delta\mathcal{D}_j^i \cap \mathcal{U}_{i-1}$, $\forall j \leq i$ and, hence, they have the same probability of being labeled or not at instant $i-1$ by any check node of any matrix $\mathbf{A}_j$ $j \leq i-1$. This fact also implies that subjects can be indexed in several sets $\Delta\mathcal{D}_j^i \cap \mathcal{U}_{i-1}$, $\forall j \leq i$, since their presence in one set depends only of the construction of $\mathbf{A}_j$, the test vector $\mathbf{y}_j$ and the subjects indexed in $\mathcal{N}_i$.

Finally, the sets $\mathcal{N}_i$ and $\mathcal{D}_i$ can be easily obtained as

$$\mathcal{N}_i \equiv \mathcal{N}_{i-1} \cup \Delta\mathcal{N}_i \qquad\qquad \mathcal{D}_i \equiv \mathcal{D}_{i-1} \cup \Delta\mathcal{D}_i \tag{6.1}$$

because by the definition both $\mathcal{N}_{i-1} \cap \Delta\mathcal{N}_i = \emptyset$ and $\mathcal{D}_{i-1} \cap \Delta\mathcal{D}_i = \emptyset$.

6.2.2 State update equations for policy design

In this section we derive all the equations required to update the parameters $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$ and $\mathrm{P}\big(\mathcal{U}_i\big)$ for Scenario 2, i.e. when no observations are available.

We assume throughout all the section that *i*) all probabilities of the previous instants j, $1 \leq j \leq i-1$ have been previously deduced, *ii*) all subjects are initially unlabeled, $\mathrm{P}(\mathcal{U}_0) = 1$ and hence initially $\mathrm{P}(\mathcal{D}_0) = 0$ and $\mathrm{P}(\mathcal{N}_0) = 0$ and *iii*) s_0 is initially known, so the probability of defective subjects $\mathrm{P}(\mathcal{D}_0^c)$ is known.

We focus on obtaining the probabilities $\mathrm{P}\big(\mathcal{D}_i\big)$ and $\mathrm{P}\big(\mathcal{N}_i\big)$, since the parameters $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$ and $\mathrm{P}\big(\mathcal{U}_i\big)$ can be derived from them as follows

$$\mathrm{P}\big(\mathcal{U}_i\big) = 1 - \mathrm{P}\big(\mathcal{D}_i\big) - \mathrm{P}\big(\mathcal{N}_i\big) \tag{6.2}$$

$$\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big) = \frac{\mathrm{P}\big(\mathcal{D}_0^c\big) - \mathrm{P}\big(\mathcal{D}_i\big)}{\mathrm{P}\big(\mathcal{U}_i\big)} \tag{6.3}$$

and $\mathrm{P}\big(\mathcal{N}_i^c \mid \mathcal{U}_i\big) + \mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big) = 1$. We obtain the terms $\mathrm{P}\big(\mathcal{D}_i\big)$ and $\mathrm{P}\big(\mathcal{N}_i\big)$ from the definitions in (6.1) as

$$\mathrm{P}\big(\mathcal{D}_i\big) = \mathrm{P}\big(\mathcal{D}_{i-1}\big) + \mathrm{P}\big(\Delta\mathcal{D}_i\big) \tag{6.4}$$

$$\mathrm{P}\big(\mathcal{N}_i\big) = \mathrm{P}\big(\mathcal{N}_{i-1}\big) + \mathrm{P}\big(\Delta\mathcal{N}_i\big) \tag{6.5}$$

where the terms $\mathrm{P}(\mathcal{D}_{i-1})$, $\mathrm{P}(\mathcal{N}_{i-1})$ and $\mathrm{P}(\mathcal{U}_{i-1})$ in equations (6.4) and (6.5) can be assumed to be known at the i-th round, since they were computed previously.

As a summary of this subsection, we summarize here the equations employed to update the parameters $\mathrm{P}(\mathcal{U}_i)$ and $\mathrm{P}(\mathcal{D}_i^c \mid \mathcal{U}_i)$ as indicated by equations (6.2) and (6.3) with the parameters $\mathrm{P}(\mathcal{N}_i)$ and $\mathrm{P}(\mathcal{D}_i)$ from equations (6.4) and (6.5). $\mathrm{P}(\mathcal{N}_i)$ results from updating equation (6.12) and applying its result to equation (6.5).

Afterwards we can update $\mathrm{P}(\mathcal{D}_i)$ in equation equations (6.4) executing, $\forall m$, $1 \leq m \leq i$, equations (6.14), (6.13), (6.15). Then also for $\forall m$, $1 \leq m \leq i$ (6.18), (6.19), (6.17), (6.20) and apply the results obtained into (6.16) and into (6.21) to obtain $\mathrm{P}(\Delta\mathcal{D}_m^i \mid \mathcal{D}_{i-1}^c)$. Finally, we can obtain $\mathrm{P}(\Delta\mathcal{D}_i)$ from equation (6.22) through the intermediate step described by equation (6.23).

Tracking the evolution of the non-defective subjects

In this subsection we show the process to derive the term $\mathrm{P}(\Delta\mathcal{N}_i)$ in equation (6.5). We exploit the graph representation of the measurement matrix to characterize this term, i.e. we *count* the amount of check nodes generated at instant i whose result is non-defective and then we count the amount of subjects that contributed to these nodes. Note that all the subjects that contributed to them are non-defective (remark 2.1), so these subjects are enclosed in the set $\mathcal{N}_i^i$.

We apply $\Delta\mathcal{N}_i = \mathcal{N}_i^i$ to equation (6.5) so

$$\mathrm{P}(\Delta\mathcal{N}_i \mid \mathcal{U}_{i-1}) = \mathrm{P}(\mathcal{N}_i^i \mid \mathcal{U}_{i-1}) \tag{6.6}$$

and hence we focus on obtaining $\mathrm{P}(\mathcal{N}_i^i \mid \mathcal{U}_{i-1})$.

As in section 4.2.1, we can obtain the fraction of subjects labeled as non-defective by the i-th round of tests because it coincides with the fraction of edges departing of tests of the i-th round that are non-defective $\mathrm{P}(y \in \mathcal{N}_y^i)$[1]. The fraction of non-defective tests depends on both the fraction of non-defective subjects at the end ofthe previous stage $\mathrm{P}\left(\mathcal{N}_{i-1}^c \mid \mathcal{U}_{i-1}\right)$ and the amount of subjects that contribute to generate each test d.

We can obtain this probability introducing the dependence with the amount of edges of the nodes, i.e. the cardinal of $|\mathcal{J}^c|$, as

$$\mathrm{P}(\mathcal{N}_y^i) = \sum_{\forall d} \mathrm{R}_i(d)\, \mathrm{P}(\mathcal{N}_y^i \mid |\mathcal{J}^c| = d) = \sum_{\forall d} \mathrm{R}_i(d)\, \mathrm{P}(|\mathcal{J}^c \cap \mathcal{N}_{i-1}^c| = d \mid |\mathcal{J}^c| = d) \tag{6.7}$$

where $\mathrm{R}_i(d)$ is the fraction of tests of the i-th stage with d contributors and $\mathrm{P}(\mathcal{N}_y^i \mid |\mathcal{J}^c| = d)$ is the probability of obtaining a non-defective test with d contributors. Due to remark 2.1

1 Note that this fraction coincides with the fraction of check nodes of type I generated at instant i in section 2.3.3.

the probability that a given test is negative is equal to the probability that all the subject contributing to it are non-defective[1]

$$\mathrm{P}\big(\mathcal{N}_y^i \;\big|\; |\mathcal{J}^c| = d\big) = \mathrm{P}\big(\mathcal{N}_{i-1}^c \;\big|\; \mathcal{U}_{i-1}\big)^d \tag{6.8}$$

We can obtain $\mathrm{P}\big(\mathcal{N}_i^i \;\big|\; \mathcal{U}_{i-1}\big)$ noting that the fraction of edges of $\mathbf{A}_i$ connected to variable nodes representing non-defective subjects at instant i equals the average number of contributors to non-defective tests divided by the average number of edges per check node, as follows

$$\mathrm{P}\big(\mathcal{N}_i^i \;\big|\; \mathcal{U}_{i-1}\big) = \frac{\sum_{\forall d\geq 0} d\,\mathrm{R}_i\,(d)\,\mathrm{P}\big(\mathcal{N}_{i-1}^c \;\big|\; \mathcal{U}_{i-1}\big)^d}{\sum_{\forall d\geq 0} d\,\mathrm{R}_i\,(d)} \tag{6.9}$$

Therefore, the term $\mathrm{P}\big(\Delta\mathcal{N}_i \;\big|\; \mathcal{U}_{i-1}\big)$ in equation (6.5) is obtained from equation (6.9) as indicated by equation (6.6).

Note that we can directly obtain $\mathrm{P}\big(\Delta\mathcal{N}_i\big)$ recalling that due to equation (5.2), the average number of edges of the graph generated at the i-th round from the point of view of variable nodes is

$$\sum_{\forall d\geq 0} d_v\,\mathrm{L}_i\,(d) = \mathrm{P}\big(\mathcal{U}_{i-1}\big) \tag{6.10}$$

where we applied that the variable node degree profile is given in equation (5.3) . Then we apply the results of equations (6.9) and (6.10) to obtain

$$\begin{aligned}\mathrm{P}\big(\Delta\mathcal{N}_i\big) &= \mathrm{P}\big(\mathcal{N}_i^i \;\big|\; \mathcal{U}_{i-1}\big)\,\mathrm{P}\big(\mathcal{U}_{i-1}\big) \\ &= \left(\sum_{\forall d\geq 0} d\,\mathrm{L}_i\,(d)\right) \frac{\sum_{\forall d\geq 0} d\,\mathrm{R}_i\,(d)\,\mathrm{P}\big(\mathcal{N}_{i-1}^c \;\big|\; \mathcal{U}_{i-1}\big)^d}{\sum_{\forall d\geq 0} d\,\mathrm{R}_i\,(d)}\end{aligned} \tag{6.11}$$

which rearranging terms reduces to a more compact expression

$$\mathrm{P}\big(\Delta\mathcal{N}_i\big) = \mathrm{r}_i \sum_{\forall d\geq 0} d\,\mathrm{R}_i\,(d)\,\mathrm{P}\big(\mathcal{N}_{i-1}^c \;\big|\; \mathcal{U}_{i-1}\big)^d \tag{6.12}$$

where r_i is the rate of matrix $\mathbf{A}_i$ as in equation (5.1).

1 This term coincides with the probability of check type I in equation (2.4) in section 2.3.3.

Tracking the evolution of the defective subjects

We devote this section to derive the equations required to characterize the evolution of the sets of defective subjects from instant $i-1$ to instant i. Specifically, here we compute the evolution of the probabilities $\mathrm{P}(\mathcal{D}_m^i \mid \mathcal{D}_{m-1}^c)$ and $\mathrm{P}(\Delta\mathcal{D}_m^i \mid \mathcal{D}_{m-1}^c)$, then we focus on obtaining a probability of $\mathrm{P}(\Delta\mathcal{D}_m^i \mid \mathcal{U}_{i-1})$ and finally $\mathrm{P}(\Delta\mathcal{D}_i)$.

First we focus on characterizing $\mathrm{P}\big(\mathcal{D}_m^i \;\big|\; \mathcal{U}_{m-1}\big)$ exploiting the graph representation of the succession of matrices $\mathrm{A}(i)$ with the counting edges strategy. We enclose here the two most relevant equations whereas the whole derivation is in appendix 6.A. So

$$\mathrm{P}\big(\mathcal{D}_m^i \;\big|\; \mathcal{U}_{m-1}\big) = \frac{\sum_{d\geq 0} \mathrm{R}_m(d)\, d\, \mathrm{P}\big(\mathcal{D}_{m-1}^c \,\big|\mathcal{U}_{m-1}\big)\big(\mathrm{P}\big(\mathcal{N}_{m-1}^c \,\big|\mathcal{U}_{m-1}\big)\,\mathrm{P}\big(\mathcal{N}_i \,\big|\mathcal{N}_m^c\big)\big)^{d-1}}{\sum_{d\geq 0} d\,\mathrm{R}_m(d)} \tag{6.13}$$

where $\mathrm{R}_m(d)$ is the check node degree profile of matrix $\mathbf{A}_m$, the terms $\mathrm{P}\big(\mathcal{N}_{m-1}^c \;\big|\; \mathcal{U}_{m-1}\big)$ and $\mathrm{P}\big(\mathcal{D}_{m-1}^c \;\big|\; \mathcal{U}_{m-1}\big)$ are already available at instant i because they were computed at instant $m < i$ and the term $\mathrm{P}\big(\mathcal{N}_i \mid \mathcal{N}_m^c\big)$ can be obtained as follows

$$\mathrm{P}\big(\mathcal{N}_i \mid \mathcal{N}_m^c\big) = \frac{\sum_{j=m+1}^{i} \mathrm{P}(\Delta\mathcal{N}_j)}{\mathrm{P}(\mathcal{N}_0^c) - \sum_{j=1}^{m} \mathrm{P}(\Delta\mathcal{N}_j)} \tag{6.14}$$

Next we can easily obtain $\mathrm{P}\big(\Delta\mathcal{D}_m^i \mid \mathcal{D}_{m-1}^c\big)$, $m \leq i$, as

$$\mathrm{P}\big(\Delta\mathcal{D}_m^i \mid \mathcal{D}_{m-1}^c\big) = \frac{\mathrm{P}\big(\mathcal{D}_m^i \mid \mathcal{U}_{m-1}\big) - \mathrm{P}\big(\mathcal{D}_m^{i-1} \mid \mathcal{U}_{m-1}\big)}{\mathrm{P}\big(\mathcal{D}_{m-1}^c \mid \mathcal{U}_{m-1}\big)} \tag{6.15}$$

where we applied the Bayes rule and that $\mathcal{D}_m^{i-1} \subseteq \mathcal{D}_m^i$[1] holds, as stated before in section 6.2.1.

The next step is to obtain $\mathrm{P}\big(\Delta\mathcal{D}_m^i \mid \mathcal{D}_{i-1}^c\big)$. With this goal in mind we first focus on computing $\mathrm{P}\big(\Delta\mathcal{D}_m^i \cap \mathcal{D}_{i-1}^c \mid \mathcal{D}_{m-1}^c \backslash \mathcal{D}_m^{i-1}\big)$. It holds that

$$\mathrm{P}\big(\Delta\mathcal{D}_m^i \cap \mathcal{D}_{i-1}^c \mid \mathcal{D}_{m-1}^c \backslash \mathcal{D}_m^{i-1}\big) = \mathrm{P}\big(\Delta\mathcal{D}_m^i \,\big|\, \mathcal{D}_{m-1}^c \backslash \mathcal{D}_m^{i-1}\big)\,\mathrm{P}\big(\mathcal{D}_{i-1}^c \,\big|\, \mathcal{D}_{m-1}^c \backslash \mathcal{D}_m^{i-1}\big) \tag{6.16}$$

because all the subjects indexed in $\Delta\mathcal{D}_m^i$ are labeled by matrices independently picked at random from different ensembles, so at instant i all subjects labeled by any stage m are selected independently and at random from the set $\mathcal{D}_{m-1}^c \backslash \mathcal{D}_m^{i-1}$.

1 Note that $\mathcal{D}_i^{i-1} = \emptyset$ because no subjects can be labeled by a stage of tests before being generated, so $\mathrm{P}\big(\mathcal{D}_i^{i-1} \mid \mathcal{D}_{i-1}^c\big) = 0$.

First we compute $\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{U}_{m-1}\big)$ as follows

$$\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{U}_{m-1}\big) = \mathrm{P}\big(\mathcal{D}^c_{m-1} \mid \mathcal{U}_{m-1}\big) - \mathrm{P}\big(\mathcal{D}^{i-1}_m \mid \mathcal{U}_{m-1}\big) \tag{6.17}$$

$$\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{D}^c_{m-1}\big) = \frac{\mathrm{P}\big(\mathcal{D}^c_{m-1} \mid \mathcal{U}_{m-1}\big) - \mathrm{P}\big(\mathcal{D}^{i-1}_m \mid \mathcal{U}_{m-1}\big)}{\mathrm{P}\big(\mathcal{D}^c_{m-1} \mid \mathcal{U}_{m-1}\big)} = 1 - \mathrm{P}\big(\mathcal{D}^{i-1}_m \mid \mathcal{D}^c_{m-1}\big) \tag{6.18}$$

and then we can obtain $\mathrm{P}\big(\Delta\mathcal{D}^i_m \mid \mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m\big)$ as follows

$$\mathrm{P}\big(\Delta\mathcal{D}^i_m \mid \mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m\big) = \frac{\mathrm{P}\big(\Delta\mathcal{D}^i_m \mid \mathcal{D}^c_{m-1}\big)}{\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{D}^c_{m-1}\big)} \tag{6.19}$$

$$\mathrm{P}\big(\mathcal{D}^c_{i-1} \mid \mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m\big) = \frac{\mathrm{P}\big(\mathcal{D}^c_{i-1} \mid \mathcal{U}_{m-1}\big)}{\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{U}_{m-1}\big)} \tag{6.20}$$

where the probabilities in the numerator and denominator of equation (6.19) are given in equations (6.15) and (6.17), respectively, and the probability in the numerator of equation (6.20) was computed in a previous instant and the one in the denominator is in equation (6.18).

Then obtain the probability

$$\begin{aligned}\mathrm{P}\big(\Delta\mathcal{D}^i_m \cap \mathcal{D}^c_{i-1} \mid \mathcal{D}^c_{m-1}\big) &= \mathrm{P}\big(\Delta\mathcal{D}^i_m \cap \mathcal{D}^c_{i-1} \mid \mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m\big)\,\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{D}^c_{m-1}\big)\\ \mathrm{P}\big(\Delta\mathcal{D}^i_m \mid \mathcal{D}^c_{i-1}\big) &= \frac{\mathrm{P}\big(\Delta\mathcal{D}^i_m \cap \mathcal{D}^c_{i-1} \mid \mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m\big)\,\mathrm{P}\big(\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \mid \mathcal{D}^c_{m-1}\big)\,\mathrm{P}(\mathcal{D}^c_{m-1})}{\mathrm{P}(\mathcal{D}^c_{i-1})}\end{aligned} \tag{6.21}$$

applying Bayes and both that $\mathcal{D}^c_{i-1} \subseteq \mathcal{D}^c_{m-1} \setminus \mathcal{D}^{i-1}_m$ and that $\mathcal{D}^c_{m-1}\backslash\mathcal{D}^{i-1}_m \subseteq \mathcal{D}^c_{m-1}$.

We can assume that the subjects indexed in $\Delta\mathcal{D}^i_m \cap \mathcal{D}^c_{i-1}$ are chosen independently at random among the subjects enclosed in $\mathcal{D}^c_{i-1}$ because the matrices $\mathbf{A}_m$ were also picked at random from different ensembles of matrices Γ_m, $1 \le m \le i$. Thus, we can find $\mathrm{P}\big(\Delta\mathcal{D}_i\big)$ as indicated by definition 6.7

$$\mathrm{P}\big(\Delta\mathcal{D}_i\big) = \mathrm{P}\Big(\bigcup_{1\le j\le i} \Delta\mathcal{D}^i_j \mid \mathcal{D}^c_{i-1}\Big)\,\mathrm{P}\big(\mathcal{D}^c_{i-1}\big) \tag{6.22}$$

where we can obtain the probability of the union of the sets with the well known formula of the sum of the probabilities of two independent random variables applying next equation recursively

$$\mathrm{P}\Big(\bigcup_{1\le j\le m} \Delta\mathcal{D}^i_j \mid \mathcal{D}^c_{i-1}\Big) = \mathrm{P}\big(\Delta\mathcal{D}^i_m \mid \mathcal{D}^c_{i-1}\big) + \mathrm{P}\Big(\bigcup_{1\le j\le m-1} \Delta\mathcal{D}^i_j \mid \mathcal{D}^c_{i-1}\Big)\,\big(1 - \mathrm{P}\big(\Delta\mathcal{D}^i_m \mid \mathcal{D}^c_{i-1}\big)\big) \tag{6.23}$$

6.2.3 State update equations for 1-step prediction

Here we extend the equations in the previous section to the case where the agent has access to $\mathrm{A}(i-1)$ and $\mathrm{y}(i-1)$ and wishes to predict the state evolution if it chooses a matrix of the ensemble Γ_i (i.e. scenario 1 in section 6.1.3). In this case the agent can exploit this knowledge to produce more accurate predictions of the outcome of an action Γ_i.

At instant $i-1$ (i.e. after decoding $\mathrm{g}_e(\mathrm{A}(i-1), \mathrm{y}(i-1))$) the agent gains access to the following data sets:

- The set of subjects that remain unlabeled at the end of the $i-1$-th decoding step, $\mathcal{U}_{i-1}$.
- The set of non-defective subjects labeled by $\mathbf{A}_{i-1}$ and $\mathbf{y}_{i-1}$, $\mathcal{N}_{i-1}^{i-1}$ as described in definition 6.2.
- The sets of defective subjects labeled by $\mathbf{A}_m$ and $\mathbf{y}_m$ when the set of non-defective subjects $\mathcal{N}_{i-1}$ is taken into account, $\mathcal{D}_m^{i-1}$, as described in definition 6.1, $0 < m \leq j \leq i-1$.

So, the agent can use this new data to obtain the values of the following variables

$\mathcal{U}_{i-1}$*:* The agent can count the amount of unlabeled subjects at the end of the decoding stage $|\mathcal{U}_{i-1}|$, so it can compute the true value of

$$\mathrm{P}(\mathcal{U}_{i-1}) = |\mathcal{U}_{i-1}|/\mathrm{N} \tag{6.24}$$

instead of predicting it.

$\Delta\mathcal{N}_{i-1}$*:* (See definition 6.6) This set coincides with $\mathcal{N}_{i-1}^{i-1}$, so the agent can replace the predicted value in equation (6.12) by the true value

$$\mathrm{P}(\Delta\mathcal{N}_{i-1}) = |\mathcal{N}_{i-1}^{i-1}|/\mathrm{N}. \tag{6.25}$$

$\mathcal{D}_m^{i-1}$*:* (See definition 6.1) The agent can infer the subjects that belong to this set counting the defective subjects labeled by the check nodes of matrix $\mathbf{A}_i$ with defective tests $\mathbf{y}_i$ at the end of the $i-1$ decoding step, since the real $\mathrm{P}\big(\mathcal{D}_m^{i-1} \mid \mathcal{U}_{m-1}\big)$ in equation (6.13) is

$$\mathrm{P}\big(\mathcal{D}_m^{i-1} \mid \mathcal{U}_{m-1}\big) = |\mathcal{D}_m^{i-1}|/|\mathcal{U}_{m-1}|. \tag{6.26}$$

Hence, after evaluating these values from the real system, the agent would estimate the next state, $\mathrm{P}\big(\mathcal{U}_i\big)$ and $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$, applying equations (6.2) and (6.3) with the parameters $\mathrm{P}\big(\mathcal{N}_i\big)$ and

$\mathrm{P}\big(\mathcal{D}_i\big)$ from equations (6.4) and (6.5). $\mathrm{P}(\mathcal{N}_i)$ results from updating for $1 \leq m \leq i-1$ the term in equation (6.25) and computing equation (6.12) and applying these results to equation (6.5).

Afterwards we can update $\mathrm{P}(\mathcal{D}_i)$ in equation equations (6.4) executing, $\forall m$, $1 \leq m \leq i$, equations (6.14)[1], (6.26), (6.15). Then also for $\forall m$, $1 \leq m \leq i$ (6.18), (6.19), (6.17), (6.20) and apply the results obtained into (6.16) and into (6.21). Finally, we can obtain $\mathrm{P}(\Delta\mathcal{D}_i)$ from equation (6.22) through the intermediate step described by equation (6.23).

6.3 Validation plots

In this section we assess the accuracy of the predictions obtained using the equations presented in section 6.2 for several population sizes and probabilities of defective. We compare the performance of the adaptive scheme proposed in chapter 5, obtained by the Monte Carlo (MC) simulation with the performance predicted by the equations deduced in section 6.2.2.

The results obtained show that we can predict the evolution of the mean of $\mathrm{P}(S_i \mid \Gamma(i); s_0)$ in the N-finite regime, i.e. the evolution of $\mathrm{P}\big(\mathcal{U}_i \mid \Gamma(i); s_0\big)$ and $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i; \Gamma(i); s_0\big)$ as i increases, since it coincides with the predictions in the N-asymptotic regime.

In this section all the simulation results were obtained employing the policy in appendix 5.A. With this prefixed policy the ensemble Γ_{i+1} depends on the value of $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$. For the sake of clarity and comparison in this section we refer to the initial conditions of the simulations to as population, i.e. defined by K and N in the N finite case, simply as $\left\{\frac{|\mathcal{P}_D|}{\mathrm{N}},\mathrm{N}\right\}$. Throughout this section we assume that K is known.

Throughout all this section each realization of the population was chosen at random among all the possible realizations of the populations $\{\mathrm{P}(\mathcal{P}_D),\mathrm{N}\}$, e.g. if $\{0.0966,\mathrm{N}\}$ there are exactly 9666 and 966 defective subjects in each realization when $\mathrm{N} = 100000$ and $\mathrm{N} = 10000$, respectively.

Figure 6.1 compares the accuracy of the prediction equations with the performance of the proposed scheme obtained by MC simulation when the population $\{0.0966,\mathrm{N}\}$ with $\mathrm{N} = 10000$ and $\mathrm{N} = 100000$. These populations were obtained by averaging 10000 and 1000 realizations, respectively.

In figure 6.1 we compare the evolution of $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$ (dot-dashed line), $\mathrm{P}\big(\mathcal{U}_i\big)$ (solid line) and the average aggregated rate of the matrix (dashed line) generated up to an instant i. The horizontal axis indicates the instant i. We point out that the plots have two vertical scales to measure

1 From the terms obtained updating for $1 \leq m \leq i-1$ the term in equation (6.25) and computing equation (6.12).

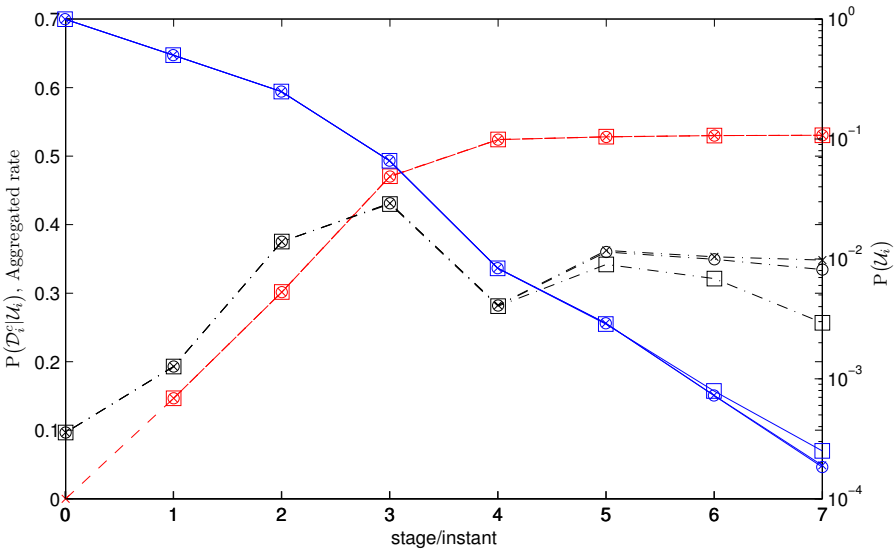


Figure 6.1: Comparison of predicted performance $N = \infty$ (marker 'x') with the one obtained by MC simulations of populations with $N = 10000$ (marker '□') and $N = 100000$ (marker 'o') of the parameters $P(\mathcal{U}_i)$ (solid blue line) and $P(\mathcal{D}_i^c \mid \mathcal{U}_i)$ (black dash-dotted line) for initial states {0.0966,N}. Figures also show the average aggregated rate of the measurement matrices (red dashed line).

the probabilities; the one on the left hand side refers to $P\left(\mathcal{D}_i^c \mid \mathcal{U}_i\right)$ and the one on the right hand side refers to $P\left(\mathcal{U}_i\right)$, e.g. the top-left plot 6.1 shows that at instant $i = 5$ $P\left(\mathcal{D}_i^c \mid \mathcal{U}_i\right)$ is approximately equal to 0.2936 and $P\left(\mathcal{U}_i\right)$ is equal to 0.03079. The average aggregated rate is referred also to the left-hand side scale and so the rate at instant $i = 5$ depicted by figure 6.1 is equal to 0.07193.

Figure 6.1 shows that the predicted evolution of $P(\mathcal{U}_i)$ and $P(\mathcal{D}_i^c \mid \mathcal{U}_i)$ matches the evolution of the parameters obtained by the MC simulation whenever the average number of subjects that remain to be labeled is large enough, since for instant $i \leq 4$ the plots representing the evolution of $P\left(\mathcal{U}_i\right)$, $P\left(\mathcal{D}_i^c \mid \mathcal{U}_i\right)$ and the average aggregated rate coincide both the ones obtained analytically and the ones obtained by MC simulations.

A mismatch between the analytical results and the ones obtained by MC simulation arises for $i \geq 5$. Initially the predicted and the simulated $P\left(\mathcal{D}_i^c \mid \mathcal{U}_i\right)$ start to diverge and then, due to this divergence at later instant i the plots $P\left(\mathcal{U}_i\right)$ start to diverge. The plots representing the MC simulation with the N (depicted by squares) diverges earlier from the predicted behavior that the ones representing the larger population (depicted by circles).

This effect can be attributed to the fact that in average only a few subjects remain unlabeled at the later stages of the testing process, e.g. for $N = 10^4$ the average amount of subjects in $\mathcal{U}_5$ and

$\mathcal{U}_6$ are equal to 28.6 and 7.9 respectively, so the assumption that the amount of subjects is large enough, made throughout all the chapter, does not hold.

In spite of this inaccuracy both the predicted and simulated average aggregated rate of the measurement matrices coincide, since in the later stages of the testing process the amount of generated test is low in comparison to the amount of generated tests in the early stages of the process. We would like to add that all simulations stopped once all the subjects were labeled. We decided to omit the result for instant i larger than the ones depicted in the figures because the amount of realizations of the simulation that achieved those instants was too low to obtain relevant results.

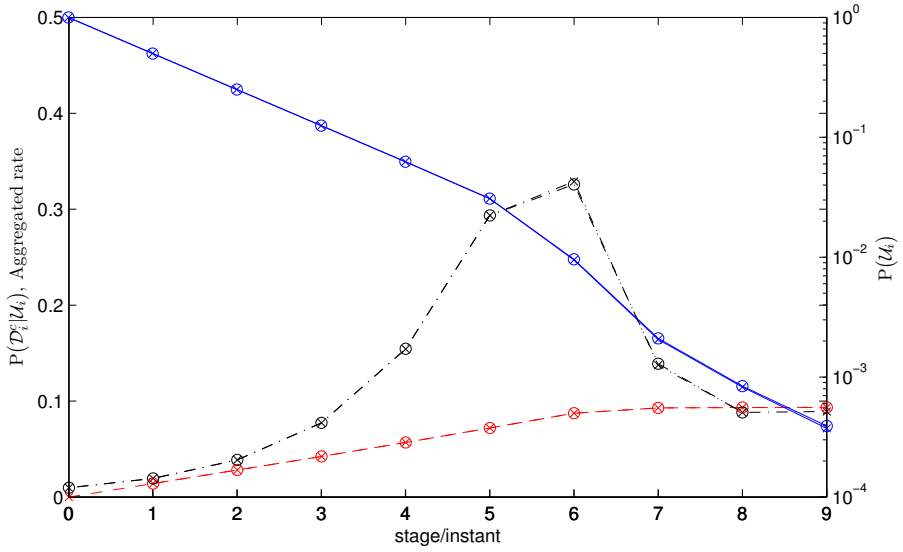


Figure 6.2: Comparison of predicted performance $\mathrm{N} = \infty$ ('x') with MC simulations of $\mathrm{N} = 100000$ ('o') of the parameters $\mathrm{P}(\mathcal{U}_i)$ (solid blue line) and $\mathrm{P}(\mathcal{D}_i^c \mid \mathcal{U}_i)$ (black dot point line) for different initial state $\{0.00966\}$. Figure also show the theoretical average aggregated rate (red striped line).

Figure 6.2, 6.3 and 6.4 compare the predicted and the simulated evolution of the parameters for several populations and as opposed to figure 6.1 they focus on the larger population sizes, $\mathrm{N} = 100000$ ($\{0.00966, \mathrm{N}\}$, $\{0.05660, \mathrm{N}\}$ and $\{0.12660, \mathrm{N}\}$ in figures 6.2, 6.3 and 6.4 respectively). Each simulation was obtained by averaging the results obtained from 1000 MC simulations whose realizations were randomly picked among all the possible realizations of the population.

Figures 6.2, 6.3 and 6.4 show that it is possible to predict accurately the evolution of the parameters $\mathrm{P}\big(\mathcal{D}_i^c \mid \mathcal{U}_i\big)$ and $\mathrm{P}\big(\mathcal{U}_i\big)$ obtained by MC simulations. In this case all plots show that the predictions obtained from the equations to track the parameters were accurate.

Figures 6.2 and 6.4 show that the prediction accurately describes both populations with a large and a reduced fraction of defective subjects. In figures 6.3 and 6.4 we omitted the results obtained

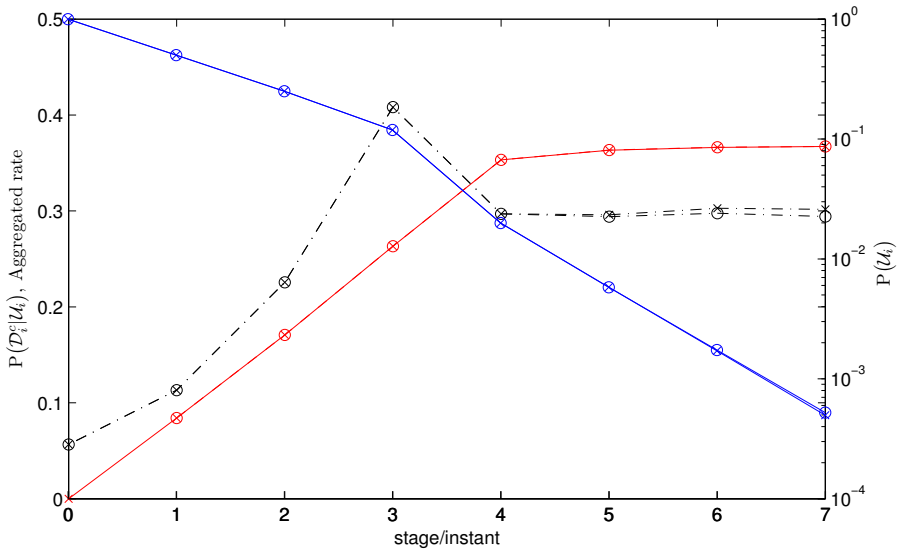


Figure 6.3: Comparison of predicted performance $N = \infty$ ('x') with MC simulations of $N = 100000$ ('o') of the parameters $P(\mathcal{U}_i)$ (solid blue line) and $P(\mathcal{D}_i^c \mid \mathcal{U}_i)$ (black dot point line) for different initial state $\{0.0566\}$. Figure also show the theoretical average aggregated rate (red striped line).

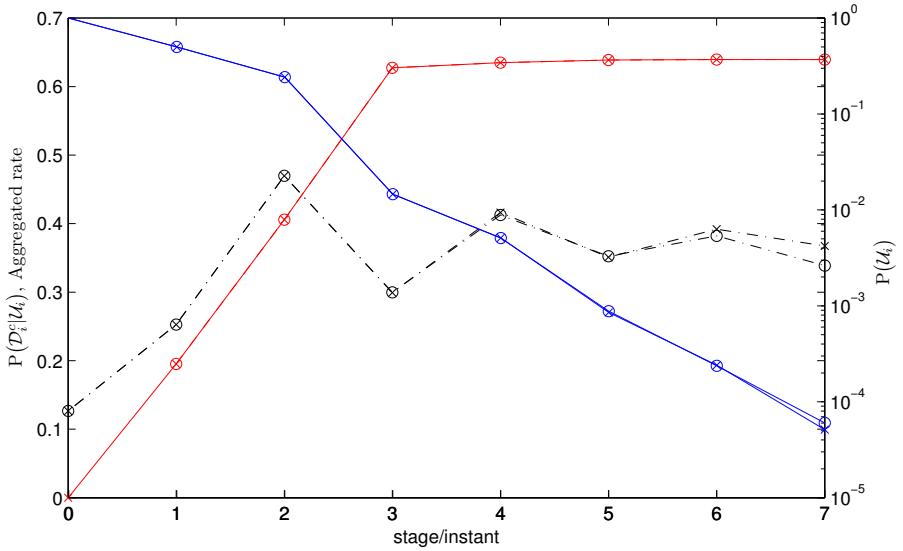


Figure 6.4: Comparison of predicted performance $N = \infty$ ('x') with MC simulations of $N = 100000$ ('o') of the parameters $P(\mathcal{U}_i)$ (solid blue line) and $P(\mathcal{D}_i^c \mid \mathcal{U}_i)$ (black dot point line) for different initial state $\{0.1266\}$. Figure also show the theoretical average aggregated rate (red striped line).

for instants $i \geq 7$, since the plots of figures started to diverge as shown in 6.4.

We included results for $i = 8{,}9$ in figure 6.2 to depict a different behavior of the simulator. Note that the plot that depicts the evolution of parameter $P\left(\mathcal{D}_i^c \mid \mathcal{U}_i\right)$ has no value at instant $i = 9$.

This happened because only unlabeled subjects remained to be labeled at the end of stage $i = 9$ in all realizations that required nine stages.

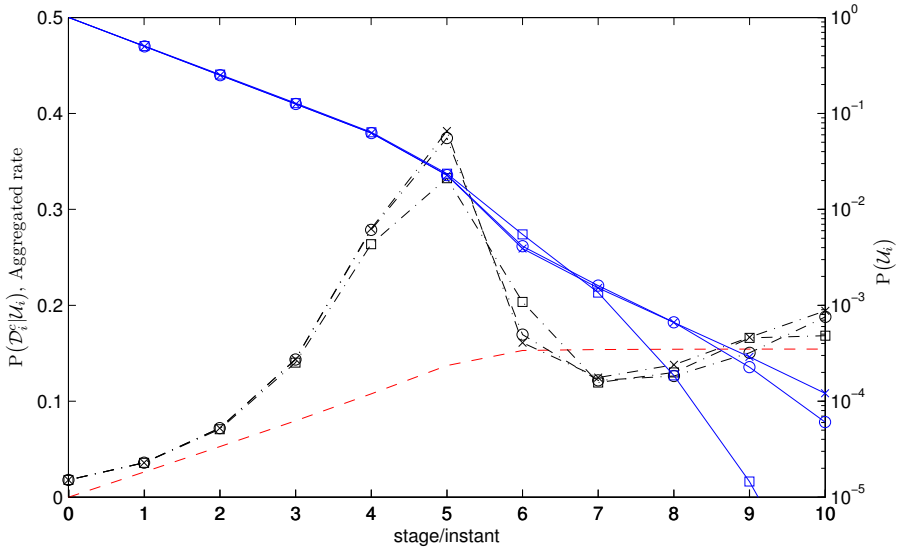


Figure 6.5: Comparison of predicted performance $N = \infty$ (x) with MC simulations of population $\{0.0180, N\}$ with $N = 1000$ (□) and $N = 10000$ (o) of the parameters $P(\mathcal{U}_i)$ (solid blue line) and $P(\mathcal{D}_i^c \mid \mathcal{U}_i)$ (black dot point line) for different populations. Figures also show the theoretical average aggregated rate (red striped line).

Figures 6.6 and 6.5 focus on comparing the predicted averages of the parameters $P(\mathcal{U}_i \mid \Gamma(i),s_0)$ and $P(\mathcal{D}_i^c \mid \Gamma(i),s_0)$ with the actual $P(\mathcal{U}_i \mid A(i), y(i),s_0)$ obtained by MC of the populations $\{0.0180, N\}$ with $N = 1000$ (□) and $N = 10000$. For the sake of comparison we added a figure to present the results as for figures 6.1, 6.2, 6.3, 6.4, 6.6 and 6.5. Figures 6.6 and 6.5 were obtained by averaging the results of 780000 and 3000000 simulations for $N = 10000$ and $N = 1000$, respectively, and each realization was randomly piked among all the realizations of the population.

Figure 6.6 depicts the shapes of the estimated probability distribution in the MC simulations and a vertical line for the predicted parameter for $i = 1,\ldots,5$. The solid and the dash-dotted lines represent evolution for instants $i = 1,\ldots,5$ of the probability distribution $P(\mathcal{U}_i \mid A(i), y(i),s_0)$ $N = 10^4$ and $N = 10^3$, respectively. For the sake of comparison we depict the normalized distribution (the horizontal axis is normalized by dividing by N).

Figure 6.6 shows that the distribution for $N = 10^4$ initially is close to the bell of a binomial distribution whereas the one with $N = 10^3$ is multi modal. The figure also shows that in both cases the probability mass concentrates around the predicted values. Furthermore, it also shows that, irrespective of the amount of subject of the population as i increases the distributions become less and less scattered, concentrating more and more around the predicted value of its mean.

Figure 6.5 shows that the prediction of $\mathrm{P}(\mathcal{U}_i)$ is initially accurate and becomes more and more rough as time passes. The estimated parameters of the simulation match the predicted behavior of the parameters better than the ones of the simulation with a reduced population. Note that behaviour of the scheme for $\mathrm{N} = 1000$ is accurately predicted even though the distributions in in figure 6.6 are not binomial.

Note also that for $i = 7$ the average number of unlabeled subjects of the simulation with $\mathrm{N} = 10^3$ is close to one, so most simulations do not reach this instant. It must be remarked that even though that the prediction with $\mathrm{N} = 10^3$ of the fraction of defective subjects that remain unlabeled at i is rough from early stages it also keeps close to the simulated behavior of the scheme irrespective of the instant. It must be also noted that the fraction of unlabeled subjects deviates from the predicted behavior falling abruptly. For $\mathrm{N} = 10^4$ the evolution of the parameters obtained from the simulation matches its predicted behavior.

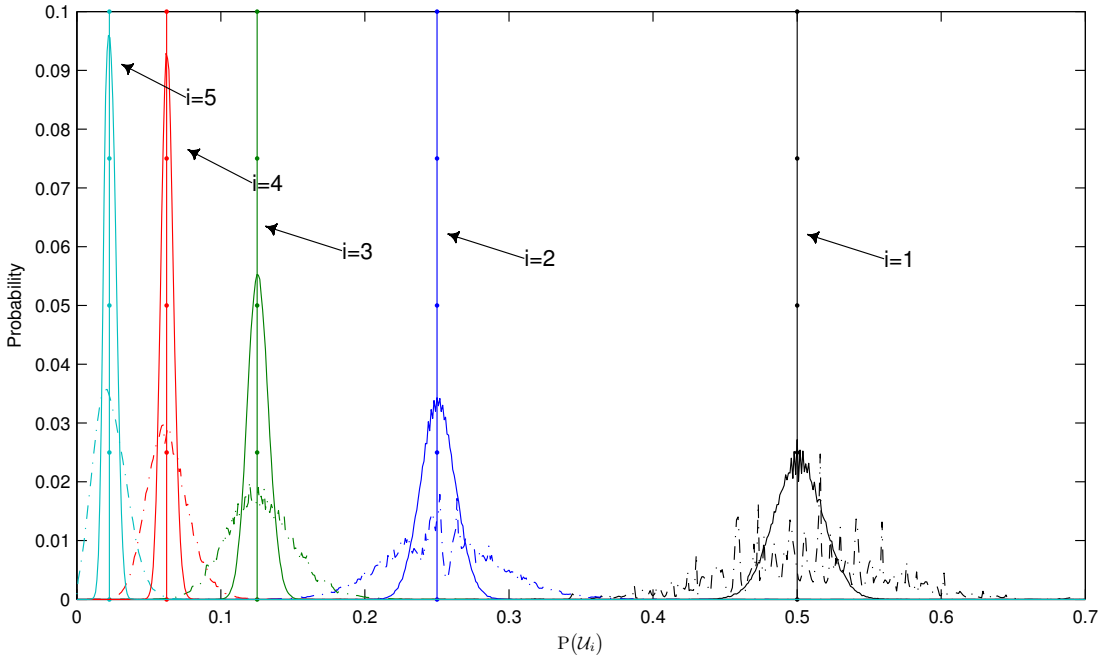


Figure 6.6: Comparison of the probability mass function $\mathrm{P}(\mathcal{U}_i \mid \mathrm{A}(i), \mathrm{y}(i), s_0)$ obtain by MC of population $\{0.0180, \mathrm{N}\}$ $\mathrm{N} = 1000$ (dashed line) $\mathrm{N} = 10000$ (solid line) of the parameters $\mathrm{P}(\mathcal{U}_i)$ with the predicted average of $\mathrm{P}(\mathcal{U}_i \mid \Gamma(i), s_0)$ (vertical lines).

Appendix 6.A Proof

Let us focus on $\mathrm{P}\Big(\mathcal{D}_m^i \;\Big|\; \mathcal{U}_{m-1}\Big)$. We can apply counting edges strategy, as in equation (6.9) in and label one defective subject per defective test where all but one subject are already labeled, i.e. a test node with all but one subject labeled as non-defective

$$\mathrm{P}\Big(\mathcal{D}_m^i \;\Big|\; \mathcal{U}_{m-1}\Big) = \frac{\sum_{d\geq 0}\; \mathrm{R}_m\,(d)\,\mathrm{P}\Big(|\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1, |\mathcal{I}\cap\mathcal{N}_i| = |\mathcal{I}|-1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}| = d\Big)}{\sum_{d\geq 0} d\,\mathrm{R}_m\,(d)} \tag{6.27}$$

where $|\mathcal{I}\cap\mathcal{D}_{m_1}^c| = 1$ represents the test nodes of the graph representation of $\mathbf{A}_m$ that have in their neighborhood $\mathcal{I}$ the contribution of a single defective subject indexed in $\mathcal{D}_{m-1}^c$ and $|\mathcal{I}\cap\mathcal{N}_i| = |\mathcal{I}|-1$ indicates that all but one subjects in the neighborhood of the test nodes are already labeled as non-defective at the i-th round. We can model the probability $\mathrm{P}\Big(|\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1, |\mathcal{I}\cap\mathcal{N}_i| = |\mathcal{I}|-1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}| = d\Big)$ in equation (6.27) as follows

$$\begin{aligned}\mathrm{P}\Big(|\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1, |\mathcal{I}\cap\mathcal{N}_i| = |\mathcal{I}|-1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}| = d\Big) &= \mathrm{P}\Big(|\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}| = d\Big)\times \\ &\times \mathrm{P}\Big(|\mathcal{I}\cap\mathcal{N}_i| = d-1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1, |\mathcal{I}| = d\Big)\end{aligned} \tag{6.28}$$

where we can model the first term in equation (6.29) with a binomial distribution in case that we assume that the amount in subjects in $\mathcal{U}_{m-1}$ is large enough and that the probability of selecting any subject to contribute to any tests is identically and independently distributed. In this situation we can write

$$\mathrm{P}\Big(|\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}| = d\Big) = \binom{d}{1}\mathrm{P}\Big(\mathcal{D}_{m-1}^c \;\Big|\; \mathcal{U}_{m-1}\Big)\;\mathrm{P}\Big(\mathcal{N}_{m-1}^c \;\Big|\; \mathcal{U}_{m-1}\Big)^{d-1} \tag{6.29}$$

The second term on the right-hand side in equation (6.28) can be written as follows

$$\mathrm{P}\Big(|\mathcal{I}\cap\mathcal{N}_i| = d-1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1, |\mathcal{I}| = d\Big) = \mathrm{P}\Big(\mathcal{N}_i \;\Big|\; \mathcal{U}_{m-1}, \mathcal{U}_m, \mathcal{N}_{m-1}^c\Big)^{d-1} \tag{6.30}$$

$\mathcal{U}_{m-1}$ implies that all the subjects were unlabeled at $m-1$ and $|\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1$ implies that one and only one was a defective subject so the other must be non-defective that remained unlabeled because each subjects contributes at most to one test at each instant. These sets reduces to $\mathcal{U}_{m-1}\cap\mathcal{U}_m\cap\mathcal{N}_{m-1}^c = \mathcal{N}_m^c$ and hence the term in equation (6.30) reduces to

$$\mathrm{P}\Big(|\mathcal{I}\cap\mathcal{N}_i| = d-1 \;\Big|\; \mathcal{U}_{m-1}, |\mathcal{I}\cap\mathcal{D}_{m-1}^c| = 1, |\mathcal{I}| = d\Big) = \mathrm{P}\Big(\mathcal{N}_i \;\Big|\; \mathcal{N}_m^c\Big)^{d-1} \tag{6.31}$$

The term $\mathrm{R}_m\,(d)$ in equation (6.27) and $\mathrm{P}\Big(\mathcal{D}_{m-1}^c \;\Big|\; \mathcal{U}_{m-1}\Big)$ in equation (6.29) can be assumed

to be already available at instant i so it only remains as unknown the term $\mathrm{P}\left(\mathcal{N}_i \;\middle|\; \mathcal{N}_m^c\right)$ in equation (6.31). This term can obtained as follows

$$\mathrm{P}\left(\mathcal{N}_i \;\middle|\; \mathcal{N}_m^c\right) = \frac{\sum_{j=m+1}^{i} \mathrm{P}(\Delta\mathcal{N}_j)}{\mathrm{P}(\mathcal{N}_0^c) - \sum_{j=1}^{m} \mathrm{P}(\Delta\mathcal{N}_j)} \tag{6.32}$$

where the terms $\mathrm{P}(\Delta\mathcal{N}_j)\ \forall j < i$ are assumed to be already computed at instant i.

PART III

Noiseless compressed sensing

CHAPTER 7

Noiseless compressed sensing limits

The noiseless compressed sensing problem deals with the reconstruction of an unknown strictly K-sparse signal in absence of noise. The K-sparse sequences are commonly referred to as the Grassmannian of K-dimensional sub-spaces of $\mathbb{R}^{\mathrm{N}}$, formally defined as $\Gamma_{\mathrm{N}}^{\mathrm{K}} \triangleq \left\{\forall \mathbf{x} \in \mathbb{R}^{\mathrm{N}} : \|\mathbf{x}\|_0 \leq \mathrm{K}\right\}$.

The noiseless analog compression limit presented in section 3.1 guarantees the existence of a noiseless CS setup composed by a linear encoder and l_0-based decoder, reviewed in section 1.1, where any sequence from Grassmanian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ can be compressed and reconstructed without losses with probability 1 whenever the rate of the scheme $r > \gamma + \mathrm{o}(1)$ for $\mathrm{K} = \mathrm{N}\gamma$. Regretfully, to the best of author's knowledge there does not exist any condition to guarantee that a matrix can achieve the theoretical compression limit.

In [135] it was shown empirically that a linear encoder with a Gaussian matrix and a MAP-based decoder with Gaussian-Bernoulli prior can achieve lossless compression of $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ for $\mathrm{M}/2 < \mathrm{K} < \mathrm{M}$. Furthermore, we also show in chapter 8 a couple more of examples of noiseless compression schemes with sparse matrices that achieve lossless reconstruction with $\mathrm{M}/2 < \mathrm{K} < \mathrm{M}$. These results indicate that there exist noiseless compression schemes that outperform the previously known compression limits for the noiseless compressed sensing problem, enclosed in lemma 3.1. Regretfully, there does not exist any sufficient condition to guarantee that a matrix enables to the l_0-based decoder to achieve lossless reconstruction for $\mathrm{M}/2 < \mathrm{K} < \mathrm{M}$.

This chapter is devoted to address this problem. Here we derive a necessary and sufficient condition to ensure that a measurement matrix $\mathbf{A}$ guarantees the lossless reconstruction of the sequences $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ with zero distortion in the N-finite regime for a non-vanishing $\gamma = \mathrm{K}/\mathrm{N}$ in noiseless CS setup.

We divide this chapter in two parts: first in section 7.1 we describe our approach to the problem and afterwards that in section 7.2 we describe the deductive process that leads to the main result: that a measurement matrix $\mathbf{A}$ with $\text{spark}(\mathbf{A}) > \mathrm{K} + 1$ guarantees that any sequence $\mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{\mathrm{K}}$ with non-null components independently chosen at random can be reconstructed with zero distortion with probability 1.

7.1 Problem statement

In this section we present our approach to pose the noiseless compressed sensing problem as an analog compression one. First, in section 7.1.1 we briefly review the notation and the compression scheme. Next, in section 7.1.2 we describe the signal characterization that we employ throughout all the chapter and compare it with the one of the analog compression setup. Finally, in section 7.1.3 we introduce our approach to the problem.

7.1.1 Scenario definition

We assume throughout all the chapter that the sequences $\mathbf{x}_0$ are chosen at random from the Grassmanian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ as stated before, i.e. first selecting a partition at random among the $\binom{\mathrm{N}}{\mathrm{K}}$ partitions and then populating the non-null coefficient with K realizations of a continuous distribution with Renyi dimension of the information 1.

Then, sequence $\mathbf{x}_0$ is processed by a linear encoder $\mathrm{q}(\cdot) : \mathcal{X} \subseteq \mathbb{R}^{\mathrm{N}} \to \mathcal{Y} \subseteq \mathbb{R}^{\mathrm{M}}$ that maps sequences $\mathbf{x}_0 \in \mathcal{X} \subset \mathbb{R}^{\mathrm{N}}$ into sequences of lower dimension $\mathbf{y}_0 \in \mathcal{Y} \subset \mathbb{R}^{\mathrm{M}}$, $\mathrm{M} < \mathrm{N}$. Afterwards a decoding function $\mathrm{g}(\cdot) : \mathcal{Y} \subseteq \mathbb{R}^{\mathrm{M}} \to \mathcal{X} \subseteq \mathbb{R}^{\mathrm{N}}$ attempts to generate $\mathbf{x} = \mathrm{g}(\mathrm{q}(\mathbf{x}_0))$, a non-distorted version of the original sequence $\mathbf{x}_0$.

We focus our attention on schemes with a linear encoder $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$ where $\mathbf{A} \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}}$ and decoders that implement the l_0 minimization problem

$$\hat{\mathbf{x}} = \min_{\mathbf{x}\in\mathcal{X}:\ \mathbf{y}_0=\mathbf{A}\mathbf{x}} \|\mathbf{x}\|_0$$

which in the ideal case can be solved by exhaustive search, with unaffordable complexity. In order to do so the decoder performs two steps: first it generates the set of solutions $\mathcal{X}(\mathbf{y}) \subseteq \mathcal{X}$ and then selects the solution $\mathbf{x}$ with minimum norm. Alternatively, if the solution is unique then the selection step can be suppressed, i.e. in this case the encoder $\mathbf{A}$ performs a one-to-one mapping.

7.1.2 Noiseless analog compression vs. classical noiseless compressed sensing. The source model

First let us compare the sequences generated by a source modeled by a mixture distribution, e.g. a Gaussian Bernoulli distribution with sparsity γ, and the sequences enclosed in the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$.

We follow an approach to model a Grassmannian similar to the one introduced by the authors in [141], i.e. a formulation that converts multidimensional signals into points on the Grassmannian manifold. This approach facilitates the study of embedding the referred manifolds via Gaussian matrices into a $\mathbb{R}^{\mathrm{M}}$ space. Through all the work, we refer to these $\binom{\mathrm{N}}{\mathrm{K}}$ K-dimensional sub-spaces embedded into the manifold $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ simply as partitions $\mathcal{V}_i$ where all the K-sparse sequences with support set $\mathcal{S}_i$ are enclosed. The basic idea is that the Grassmannian manifold can be represented as $\binom{\mathrm{N}}{\mathrm{K}}$ partitions, each one representing all the sequences of a $\mathbb{R}^{\mathrm{K}}$ space.

We could assume that sequences $\mathbf{x}_0$ of the Grassmanian are generated by a data source that, first, selects one partition among the $\binom{\mathrm{N}}{\mathrm{K}}$ possible choices with uniform probability and, second, populates the K non-null coefficients with i.i.d. random variables chosen according to some probability distribution f(X) with Renyi dimension of the information space equal to 1, e.g. a Gaussian distribution with $\sigma^2 = 1$.

From the point of view of Grassmanian manifolds, a source distributed according to the mixture distribution in equation (1.1) first selects both the K to characterize the Grassmanian and the partition where the original signal lays and afterwards, chooses the exact point in that partition. Thus, in this case the probability of selecting a sequence $\mathbf{x}_0$ of the Grassmanian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ is distributed according a binomial distribution with parameters (N,K,γ), referred to as bin(N,K,γ), whereas in our signal generation model the K is predefined.

In case that N and K are large enough the binomial distribution is approximately symmetric and concentrates around its mean, i.e. Nγ. So, like in the almost lossless binary compression problem, analog compression schemes can be designed to deal exclusively with the most likely sequences (or partitions) since the unlikely sequences (partitions) may appear with vanishing probability as N grows. Hence, in this case compression schemes can focus on the typical sequences (partitions).

Problems arise in the N finite regime, when the error introduced by the queues of the binomial distribution cannot be obviated, since the queues of the distribution may affect significantly to the performance of the decoding scheme, producing a significant (non-vanishing) error rate.

In the N finite regime it makes sense to study the largest Grassmanian, i.e. the largest K for which the compression scheme is lossless, since the error rate in this case can be easily found as

$$\mathrm{P}(\mathbf{x}_0 \neq \mathrm{g}(\mathrm{q}(\mathbf{x}_0))) = \sum_{k=\mathrm{K}+1}^{\mathrm{N}} \mathrm{P}(\mathbf{x}_0 \neq \mathbf{x} \mid \mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{k})\, \mathrm{P}(\mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{k})$$

where $\mathbf{x}_0 = \mathrm{g}(\mathrm{q}(\mathbf{x}_0))$ for $\mathrm{k} < \mathrm{K}$ since it is straightforward that $\Gamma_{\mathrm{N}}^{k} \subset \Gamma_{\mathrm{N}}^{\mathrm{K}}$ holds for $\mathrm{k} < \mathrm{K}$. Note that terms $\mathrm{P}(\mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{k})$ are the bin(N,K,γ). Hence, we can upper bound the error rate of the compression scheme as

$$\mathrm{P}(\mathbf{x}_0 \neq \mathrm{g}(\mathrm{q}(\mathbf{x}_0))) \leq \sum_{k=\mathrm{K}+1}^{\mathrm{N}} \mathrm{P}(\mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{k}) \tag{7.1}$$

7.1.3 Approach to the problem

As stated before, we follow an approach that divides the Grassmanian manifold $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ into $\binom{\mathrm{N}}{\mathrm{K}}$ partitions $\mathcal{V}_i$, so

$$\Gamma_{\mathrm{N}}^{\mathrm{K}} \triangleq \bigcup_{\forall i} \mathcal{V}_i \tag{7.2}$$

This decomposition of $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ allows us to analyze how any $\mathcal{V}_i$ partition is projected into the $\mathbb{R}^{\mathrm{M}}$ space by the columns of matrix $\mathbf{A}$ indexed in $\mathcal{S}_i$, referred to as $\mathbf{A}_i$. Furthermore, it is also helpful to gather knowledge about how the co-domains of any two partitions i and j, referred to as $\mathrm{q}(\mathcal{V}_i)$ and $\mathrm{q}(\mathcal{V}_j)$, intersect when the mapping is non-injective. The latter approach is quite useful because it allows us to draw conclusions in an alternative and quite straightforward manner. For example, from this point of view lemma 3.1 is obvious.

Proof. Consider that $\mathcal{V}_i^{\mathrm{K}}$ and $\mathcal{V}_j^{\mathrm{K}}$ denote[1] any two different partitions of the Grassmanian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$. Consider that $\mathcal{V}_v^{\mathrm{M}}$ denotes a partition of the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{M}}$, with $\mathrm{N} \geq \mathrm{M} > \mathrm{K}$ and furthermore that both $\mathcal{V}_i^{\mathrm{K}}$ and $\mathcal{V}_j^{\mathrm{K}}$ are sub-spaces of the partition $\mathcal{V}_v^{\mathrm{M}}$.

Under these conditions it is straightforward that if the relation between $\mathcal{V}_v^{\mathrm{M}}$ and its projection is injective, then the relation between either $\mathcal{V}_i^{\mathrm{K}}$ and $\mathcal{V}_j^{\mathrm{K}}$ and their respective projections will also be whenever the rank of $\mathbf{A}_v$ is M. The latter fact happens for any partition $\mathcal{V}_v^{\mathrm{M}}$ of $\Gamma_{\mathrm{N}}^{\mathrm{M}}$ whenever the spark of the measurement matrix is maximum. □

1 The superscript K indicates that the partition belongs to the Grassmanian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ whereas the superscript M indicates that the partition belongs to the Grassmanian $\Gamma_{\mathrm{N}}^{\mathrm{M}}$.

Then it can be easily deduced that

Corollary 7.1. *The sampling of any sequence* $\mathbf{x}_0 \subseteq \Gamma_{\mathrm{N}}^{\mathrm{M}/2}$ *by a linear encoder is injective whenever the spark of the measurement matrix is maximum, i.e.* $\mathrm{M}+1$.

Furthermore, whenever the spark of matrix $\mathbf{A}$ is larger than 2K there exists a one to one mapping between $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ and $\mathrm{q}(\Gamma_{\mathrm{N}}^{\mathrm{K}})$.

We denote that a function performs a one-to-one mapping between a set and its image set as[1]

$$|\mathcal{X}| = |\mathrm{q}(\mathcal{X})|$$

and that the mapping is non-injective as

$$|\mathcal{X}| > |\mathrm{q}(\mathcal{X})|.$$

Using the latter notation, the mapping between any pair of partitions j and i and their projections generated by a linear encoder $\mathrm{q}(\cdot)$ as stated in corollary 7.1 can be written as

$$|\mathcal{V}_i \cup \mathcal{V}_j| = |\mathrm{q}\big(\mathcal{V}_i \cup \mathcal{V}_j\big)| = |\mathrm{q}\big(\mathcal{V}_i\big) \cup \mathrm{q}\big(\mathcal{V}_j\big)| \tag{7.3}$$

and it is straightforward that it holds that

$$|\mathrm{q}(\Gamma_{\mathrm{N}}^{\mathrm{K}})| = |\mathrm{q}(\bigcup_{\forall i} \mathcal{V}_i)| = |\bigcup_{\forall i} \mathrm{q}(\mathcal{V}_i)| = |\Gamma_{\mathrm{N}}^{\mathrm{K}}| \tag{7.4}$$

Let us illustrate with an example how a one-to-one encoding function $\mathrm{q}(\cdot)$ can accomplish equation (7.4) for $\mathrm{r} = \gamma + \mathrm{o}(1)$ or conversely $\mathrm{M} = \mathrm{K} + 1$:

Example 7.1 (one-to-one encoder for $\Gamma_{\mathrm{N}}^{\mathrm{K}}$)**.** *We can divide the encoding process into two parts: a first part maps* $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$ *with a real valued* $\mathrm{K} \times \mathrm{N}$ *measurement matrix with spark* $\mathrm{K}+1$*. Then we can add and an extra sample with an index 'i' to identify the partition* $\mathcal{V}_i$ *where the original sequence* $\mathbf{x}_0$ *lays.*

In the example the decoder needs only an extra sample to identify the partition where the signal lays. Similarly, a linear encoder devotes all the samples that exceed K to help to the l_0-based decoder to identify the partition where the original signal $\mathbf{x}_0$ lays.

1 We perform an extension of the notation for finite countable sets to the case when the sets are uncountable.

As stated before, it cannot be guaranteed that the CS encoder performs one-to-one mapping when $\mathrm{M}/2 < \mathrm{K} < \mathrm{M}$ since lemma 3.1 does not hold. Under this situation an arbitrary number of sequences $\mathbf{x}_i \in \mathcal{V}_i$ and $\mathbf{x}_j \in \mathcal{V}_j$, possibly infinite, with $l_0(\mathbf{x}_j - \mathbf{x}_i) \neq 0$ may be mapped to the same sequence

$$\mathbf{y} = \mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{x}_j$$

Throughout all the chapter we commit an abuse of notation because we refer to the latter expression also as[1]

$$\mathbf{y} = \mathbf{A}_i\mathbf{x}_i = \mathbf{A}_j\mathbf{x}_j \tag{7.5}$$

As stated before, when $\mathrm{M}/2 < \mathrm{K} < \mathrm{M}$ lemma 3.1 does not hold. Thus, we assume the general case: that the encoder is a non-injective function, so equation (7.3) becomes

$$|\mathcal{V}_i \cup \mathcal{V}_j| > |\mathrm{q}(\mathcal{V}_i \cup \mathcal{V}_j)| \tag{7.6}$$

We can look at it in an alternative way: when a specific compressed sequence $\mathbf{y}_0$ belongs to the intersection set $\mathrm{q}(\mathcal{V}_i) \cap \mathrm{q}(\mathcal{V}_j)$ but its original sequence $\mathbf{x}_0$ do not belong to the intersection of the partitions $\mathcal{V}_i$ and $\mathcal{V}_j$ even assuming that $|\mathcal{V}_i| = |\mathrm{q}(\mathcal{V}_i)|$ and $|\mathcal{V}_j| = |\mathrm{q}(\mathcal{V}_j)|$ holds, i.e. when the ranks of the projection matrices are maximum, $\mathrm{rank}(\mathbf{A}_i) = \mathrm{rank}(\mathbf{A}_j) = \mathrm{K}$.

We group all these compressed sequences that force the encoder to be non-injective in the set $\mathcal{Y}^e$. In parallel we can also define its complementary as $\mathcal{Y}^b \triangleq \mathrm{q}(\Gamma_{\mathrm{N}}^{\mathrm{K}}) \backslash \mathcal{Y}^e$. The latter set $\mathcal{Y}^b$ is the codomain of a subset $\mathcal{X}^b \subseteq \Gamma_{\mathrm{N}}^{\mathrm{K}}$ so as $|\mathcal{X}^b| = |\mathcal{Y}^b|$. Again, we can define $\mathcal{X}^e$ as the complementary set of $\mathcal{X}^b$ on $\Gamma_{\mathrm{N}}^{\mathrm{K}}$, $\mathcal{X}^e \triangleq \Gamma_{\mathrm{N}}^{\mathrm{K}} \backslash \mathcal{X}^b$. We refer to $\mathcal{X}^b$ as the set of injective sequences of the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ produced by $\mathbf{A}$, i.e. the ones that can be perfectly reconstructed by and optimum decoder, and alternatively $\mathcal{X}^e$ as to the set of non-injective sequences of the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ due to $\mathbf{A}$.

The main question that we address in chapter arises from equation (7.5). In general when $\mathrm{M} = \mathrm{K}$ there is a one-to-one mapping between the sequences $\mathbf{x}_i \in \mathcal{V}_i$ and $\mathbf{x}_j \in \mathcal{V}_j$ because $\mathbf{x}_i = \mathbf{A}_i^{-1}\mathbf{A}_j\mathbf{x}_j$ since the spark is $\mathrm{M}+1$ and so $\mathrm{rank}(\mathbf{A}_i) = \mathrm{M}$. Hence, $\mathcal{X}^e \equiv \Gamma_{\mathrm{N}}^{\mathrm{K}}$ in this case. However, when $\mathrm{M} > \mathrm{K}$, it becomes less likely that two sequences $\mathbf{x}_i \in \mathcal{V}_i$ and $\mathbf{x}_j \in \mathcal{V}_j$ share the same compressed sequence, i.e. the sequences $\mathbf{x}_0 \in \mathcal{X}^e \subset \Gamma_{\mathrm{N}}^{\mathrm{K}}$ become scarcer. Regretfully it cannot be ensured that no sequences share the same compressed sequence but a question arise: how unlikely are these sequences in $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ when $\mathrm{M} > \mathrm{K}$?

1 $\mathbf{x}_i$ represents both a vector of length K or N depending of the context. The reader should deduce its length according to the matrices it operates with, since $\mathbf{A}_i$ is a tall $\mathrm{M} \times \mathrm{K}$ matrix whereas $\mathbf{A}$ is a fat $\mathrm{M} \times \mathrm{N}$ matrix.

7.2 Matrix condition for lossless compression

Let us focus on the case $\mathrm{M} > \mathrm{K} > \mathrm{M}/2$, where lossless compression apparently cannot be achieved but intuitively it is likely that under some conditions several sequences $\mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{\mathrm{K}}$ are mapped one-to-one by matrix-vector product encoder within $\mathrm{q}(\Gamma_{\mathrm{N}}^{\mathrm{K}})$.

In order to obtain an injective mapping the encoder must project each partition $\mathcal{V}_i$ of the Grassmanian into $\mathbb{R}^{\mathrm{M}}$, $1 \leq i \leq \binom{\mathrm{N}}{\mathrm{K}}$, in such a manner that $\mathrm{q}\big(\mathcal{V}_i \backslash \mathcal{V}_j\big) \cap \mathrm{q}\big(\mathcal{V}_j \backslash \mathcal{V}_i\big) = \emptyset$ for all $i \neq j$, as in the example of the previous section.

Regretfully, when the encoder is linear and $\mathrm{M} > \mathrm{K} > \mathrm{M}/2$ the intersection sets are non-empty, since the support sets $\mathcal{S}_i$ and $\mathcal{S}_j$ of all the points enclosed in partitions $\mathcal{V}_i$ and $\mathcal{V}_j$ have jointly more than M components, $|\mathcal{S}_j \cup \mathcal{S}_i| > \mathrm{M}$. Therefore, in this case there exist some sequences $\mathbf{x}_i \in \mathcal{V}_i$ and $\mathbf{x}_j \in \mathcal{V}_j$ such that $\mathbf{A}\,(\mathbf{x}_i - \mathbf{x}_j) = 0$ since the number of columns vectors of matrix $\mathbf{A}$ indexed by $\mathcal{S}_j \cup \mathcal{S}_i$ is larger than M.

Note that sequence $\mathbf{x}_i - \mathbf{x}_j$ belongs to a sub-space defined by $\mathcal{V}_i \times \mathcal{V}_j$ and also to the null-space of matrix $\mathbf{A}$, denoted as $\mathcal{A}_u$ and so the intersection of both sub-spaces is non-empty. This fact indicates that the null-space of the matrix $\mathcal{A}_u$ is closely related with the sequences $\mathbf{x}_0$ of $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ that belong to $\mathcal{X}^e$.

Throughout all this section we investigate the relations between the sub-space of erroneous sequences $\mathcal{X}^e$ and the null-space of the matrix $\mathbf{A}$ with the target to obtain the value of $\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big)$.

In case that both $\mathrm{q}(\Gamma_{\mathrm{N}}^{\mathrm{K}})$ and $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ were countable, we would be able to count the number of elements in $\mathcal{X}^e$ and then obtain the fraction of compressed sequences of $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ that do not belong to $\mathcal{X}^b$ as

$$\mathrm{P}\big(\mathbf{x}_0 \in \mathcal{X}^e \;\big|\; \mathbf{x}_0 \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \frac{|\mathcal{X}^e|}{|\Gamma_{\mathrm{N}}^{\mathrm{K}}|} = \frac{|\mathcal{X}^e|}{|\mathcal{X}^b| + |\mathcal{X}^e|}. \tag{7.7}$$

As we deal with uncountable sets, we must replace the set cardinality by an appropriate measure of the set hyper-volume of both, the sub-spaces $\mathcal{X}^e$ and $\Gamma_{\mathrm{N}}^{\mathrm{K}}$.

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \frac{\mu\,(\mathcal{X}^e)}{\mu\,(\Gamma_{\mathrm{N}}^{\mathrm{K}})} \tag{7.8}$$

where $\mu(\cdot)$ is chosen to be a Lebesgue measure [76] as discussed next in section 7.2.1.

Next subsection is devoted to prove that

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \frac{\mu\left(\mathcal{X}^e\right)}{\mu\left(\Gamma_{\mathrm{N}}^{\mathrm{K}}\right)} = 0 \tag{7.9}$$

whenever $\mathrm{spark}(\mathbf{A}) > \mathrm{K}+1$ is a necessary and sufficient condition that guarantees that the error rate of the considered compression scheme in the noiseless setting is 0. Note that the fact that $\mu\left(\mathcal{X}^e\right) = 0$ when $\mathcal{X}^e$ is not empty translates into the fact that apparently it is *unlikely* that two different sequences that belong to different partitions of the Grassmannian generate two compressed sequences that coincide.

7.2.1 Main result

This subsection can be decomposed into two parts: first, we give an expression of $\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big)$ as a function of the sets of erroneous sequences that belong to each partition $\mathcal{V}_i$ $\mathcal{V}_i^e$, $\mathcal{V}_i^e \triangleq \mathcal{V}_i \cap \mathcal{X}^e$. Second, we deduce the pairwise relation of any two different sets $\mathcal{V}_i^e$ and $\mathcal{V}_j^e$. Afterwards we show that $\mathcal{X}^e$ has K-dimensional Lebesgue measure zero whereas $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ has measure larger than zero and hence it follows that the rate of errors of the compression scheme is zero.

Let us focus now on investigating the "amount" of sequences $\mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}$ that are in the set of erroneous sequences $\mathcal{X}^e \subseteq \Gamma_{\mathrm{N}}^{\mathrm{K}}$ when $\mathrm{M} > \mathrm{K} > \mathrm{M}/2$ as a function of the $\mathrm{spark}(\mathbf{A})$.

We recall that $\bigcup_{\forall i} \mathcal{V}_i = \Gamma_{\mathrm{N}}^{\mathrm{K}}$ and apply it to the left hand side of equation (7.8) and results that

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \mathrm{P}\Big(\mathbf{x} \in \bigcup_{\forall i} \mathcal{V}_i^e \;\Big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\Big) \tag{7.10}$$

where we applied that the sequences in $\mathcal{X}^e$ belong to one or more partitions of type $\mathcal{V}_i$ and so $\mathcal{V}_i^e \triangleq \mathcal{V}_i \cap \mathcal{X}^e$, as stated before. Operating as indicated in appendix 7.A we obtain

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \sum_{\forall i} \mathrm{P}\big(\mathbf{x} \in \mathcal{V}_i^e \;\big|\; \mathbf{x} \in \mathcal{V}_i\big)\, \mathrm{P}\big(\mathbf{x} \in \mathcal{V}_i \;\big|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) \tag{7.11}$$

Now we focus on the first term in the summation in eq. (7.11)

$$\mathrm{P}\Big(\mathbf{x} \in \mathcal{V}_i^e \;\Big|\; \mathbf{x} \in \mathcal{V}_i\Big) = \frac{\mu\left(\mathcal{V}_i^e\right)}{\mu\left(\mathcal{V}_i\right)} \tag{7.12}$$

and further focus our efforts to find the dependence of the value of $\mu\left(\mathcal{V}_i^e\right)$ with $\mathrm{spark}(\mathbf{A})$. The set $\mathcal{V}_i^e \triangleq \mathcal{V}_i \cap \mathcal{X}^e$ is composed by all the sequences in $\mathcal{V}_i \in \Gamma_{\mathrm{N}}^{\mathrm{K}}$ that are not injective, as stated before. In other words, $\mathcal{V}_i^e$ contains any sequence $\mathbf{x}_i \in \mathcal{V}_i$ that has a *pair* sequence $\mathbf{x}_j \in \mathcal{V}_j$ that has the

same image $\mathbf{y}$ even though that $\mathbf{x}_i \notin \mathcal{V}_j$, $\forall j \neq i$. Let us define this set of all the sequences in $\mathcal{V}_i^e$ that have a pair sequence in $\mathcal{V}_j$ $\mathcal{V}_{i\leftarrow j}^e$ as

$$\mathcal{V}_{i\leftarrow j}^e \triangleq \{\mathbf{x} \in \mathcal{V}_i : \ \mathrm{q}(\mathbf{x}) \in \mathrm{q}(\mathcal{V}_i \backslash \mathcal{V}_j) \cap \mathrm{q}(\mathcal{V}_j \backslash \mathcal{V}_i)\}$$

Then the set $\mathcal{V}_i^e$ can be partitioned in subsets defined by $\mathcal{V}_i^e = \bigcup_{\forall j} \mathcal{V}_{i\leftarrow j}^e$. Note that the term $\mathcal{V}_{i\leftarrow i}^e$ is equal to the empty set whenever $\mathrm{spark}(\mathbf{A}) > \mathrm{K}$. So, we do not have to omit i-th term in the union or summation, as it is either empty or zero. If we introduce this decomposition in equation (7.11) results

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \ \big| \ \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \sum_{\forall i} \mathrm{P}\Big(\mathbf{x} \in \bigcup_{\forall j} \mathcal{V}_{i\leftarrow j}^e \ \Big| \ \mathbf{x} \in \mathcal{V}_i\Big)\, \mathrm{P}\big(\mathbf{x} \in \mathcal{V}_i \ \big| \ \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) \tag{7.13}$$

which reduces to

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{X}^e \ \big| \ \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) \leq \sum_{\forall i} \sum_{\forall j} \mathrm{P}\big(\mathbf{x} \in \mathcal{V}_{i\leftarrow j}^e \ \big| \ \mathbf{x} \in \mathcal{V}_i\big)\, \mathrm{P}\big(\mathbf{x} \in \mathcal{V}_i \ \big| \ \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) \tag{7.14}$$

if we apply the union bound. Next we can focus our attention on the term of the right side of the inequality. Note that it holds for all $1 \leq i \leq \binom{\mathrm{N}}{\mathrm{K}}$ that

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{V}_i \ \big| \ \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\big) = \binom{\mathrm{N}}{\mathrm{K}}^{-1} \tag{7.15}$$

due to the signal generation assumption given in section 7.1.1.

We can write $\mathrm{P}\big(\mathbf{x} \in \mathcal{V}_{i\leftarrow j}^e \ \big| \ \mathbf{x} \in \mathcal{V}_i\big)$ as

$$\mathrm{P}\big(\mathbf{x} \in \mathcal{V}_{i\leftarrow j}^e \ \big| \ \mathbf{x} \in \mathcal{V}_i\big) = \frac{\mu\left(\mathcal{V}_{i\leftarrow j}^e\right)}{\mu\left(\mathcal{V}_i\right)} \tag{7.16}$$

where $\mu\left(\cdot\right)$ must be a measure to quantify both the amount of sequences in either $\mathcal{V}_i$ and $\mathcal{V}_{i\leftarrow j}^e$. In order to do so first we define a σ-algebra of the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$, referred to as $\left(\Gamma_{\mathrm{N}}^{\mathrm{K}}, \mathcal{V}\right)$ where $\mathcal{V}$ is the collection of all sub-sets (power set) of the Grassmanian manifold $\Gamma_{\mathrm{N}}^{\mathrm{K}}$. Now we must choose a Lebesgue measure where both $\mathcal{V}_i$ and $\mathcal{V}_{i\leftarrow j}^e$ are measurable and, furthermore, where $\mathcal{V}_i$ is a set with measure larger than zero. We decide to choose K-dimensional Lebesgue measure, referred to as $\mu_{\mathrm{K}}\left(\cdot\right)$, because in general the K-dimensional Lebesgue measure of any lower dimensional real space is zero whereas the K-dimensional Lebesgue measure of a sub-space $\mathcal{V}_i$ of K-dimensional real space $\mathcal{V}_i \subseteq \mathbb{R}^{\mathrm{K}}$ can be larger than zero [76], so $\mu_{\mathrm{K}}(\mathcal{V}_i) > 0$ e.g. when $\mathcal{V}_i$ is a K-dimensional rectangle with side larger than 0.

It is straightforward to show that the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ has also a non-zero K-dimensional Lebesgue measure as $\bigcup_{\forall i} \mathcal{V}_i = \Gamma_{\mathrm{N}}^{\mathrm{K}}$ and $\mathcal{V}_i$ can be assumed to be a K-dimensional rectangle with an arbitrarily large side[1]. Under this approach the union of measurable spaces is measurable with measure

$$\mu_{\mathrm{K}}\left(\mathcal{V}_i\right) = \mu_{\mathrm{K}}\left(\bigcup_{\forall i} \mathcal{V}_i\right) \leq \mu_{\mathrm{K}}\left(\Gamma_{\mathrm{N}}^{\mathrm{K}}\right)$$

as noted in [76]. Furthermore, the set $\mathcal{V}_{i\leftarrow j}^{e}$ is also measurable as it is a subset of the measurable set $\mathcal{V}_i$. Then, it follows that any set $\mathcal{V}_i^e$ is also measurable as it is the union of several measurable sets $\mathcal{V}_{i\leftarrow j}^{e}$.

Now we are ready to present the following result

Theorem 7.1. *Consider two arbitrary partitions $\mathcal{V}_i$ and $\mathcal{V}_j$, $1 \leq i < j \leq \binom{\mathrm{N}}{\mathrm{K}}$, that represent the i-th and the j-th* K*-dimensional real spaces of the Grassmannian $\Gamma_{\mathrm{N}}^{\mathrm{K}}$ and a measurement matrix $\mathbf{A} \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}}$ with* $\operatorname{spark}(\mathbf{A}) > \mathrm{K}+1$*. Consider the measurable space defined by the σ-algebra $\left(\Gamma_{\mathrm{N}}^{\mathrm{K}}, \mathcal{V}\right)$ and the* K*-dimensional Lebesgue measure $\mu_K(\cdot)$. It holds that $\mathcal{V}_{i\leftarrow j}^{e}$ has* K*-Lebesgue measure zero*

$$\mu_{\mathrm{K}}\left(\mathcal{V}_{i\leftarrow j}^{e}\right) = 0 \tag{7.17}$$

$\forall i,j$, $1 \leq i < j \leq \binom{\mathrm{N}}{\mathrm{K}}$ *and* $0 \leq \mathrm{K} < \mathrm{M} < \mathrm{N}$.

See proof in appendix 7.B.

Then it follows that

Lemma 7.1. *Given a linear encoder described by matrix $\mathbf{A}$ with* $\operatorname{spark}(\mathbf{A}) > \mathrm{M}$*. The subset $\mathcal{X}^e \subset \Gamma_{\mathrm{N}}^{\mathrm{K}}$ with $\mathrm{M} > \mathrm{K}$ happens with probability*

$$\mathrm{P}\left(\mathbf{x} \in \mathcal{X}^e \;\middle|\; \mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}\right) = 0 \tag{7.18}$$

Proof. By direct application of theorem 7.1, to equation (7.16) and this result to equation (7.14) we obtain the result in equation (7.18) □

Hence, when the spark of the measurement matrix is maximum, it holds that

1 We can alternatively define $\mathcal{V}_i$ as the space that contains all sequence $\mathbf{x} \in \Gamma_{\mathrm{N}}^{\mathrm{K}}$ with support set $\mathcal{S}_i$ so as $\mathbf{x}:\ \|\mathbf{x}\|_2 < +\infty$.

Corollary 7.2. *Consider a linear encoder described by matrix* $\mathbf{A}$ *with* $\text{spark}(\mathbf{A}) = \text{M} + 1$ *and the signal setup described in 7.1.1. The mapping between sequences of the Grassmanian manifold* $\Gamma_{\text{N}}^{\text{M}-1}$ *and its codomain* $\text{q}(\Gamma_{\text{N}}^{\text{M}-1})$ *is injective almost everywhere, i.e. with probability 1.*

The result in corollary 7.2 indicates that a random source selecting points of the Grassmanian $\Gamma_{\text{N}}^{\text{M}-1}$ as indicated in 7.1 generates sequences $\mathbf{x}_0$ that produce erroneous reconstruction but that this event happens with probability zero. So, under these restrictions a source selecting sequences of the Grassmanian $\Gamma_{\text{N}}^{\text{K}}$ as stated in section 7.1 generates injective sequences with probability 1 whenever the $\text{spark}(\mathbf{A}) > \text{K} + 1$.

Appendix 7.A Proof

Here we enclose some steps that for the sake of clarity we skipped in section 7.2.1.

We can now decompose $\Gamma_{\text{N}}^{\text{K}}$ in equation (7.10) resulting

$$\text{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\big) = \sum_{\forall j} \text{P}\big(\mathbf{x} \in \mathcal{V}_j \;\big|\; \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\big)\, \text{P}\Big(\mathbf{x} \in \bigcup_{\forall i} \mathcal{V}_i^e \;\Big|\; \mathbf{x} \in \mathcal{V}_j, \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\Big) \tag{7.19}$$

where we applied total probabilities. Take into account that $\mathcal{V}_j \subset \Gamma_{\text{N}}^{\text{K}}$ for all j,

$$\text{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\big) = \sum_{\forall j} \text{P}\big(\mathbf{x} \in \mathcal{V}_j \;\big|\; \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\big)\, \text{P}\Big(\mathbf{x} \in \bigcup_{\forall i} \mathcal{V}_i^e \;\Big|\; \mathbf{x} \in \mathcal{V}_j\Big) \tag{7.20}$$

Now, let us focus to analyze the term $\text{P}(\mathbf{x} \in \bigcup_{\forall i} \mathcal{V}_i^e \mid \mathbf{x} \in \mathcal{V}_j)$ for a given j

$$\text{P}\Big(\mathbf{x} \in \bigcup_{\forall i} \mathcal{V}_i^e \;\Big|\; \mathbf{x} \in \mathcal{V}_j\Big) = \text{P}\big(\mathbf{x} \in \mathcal{V}_j^e \;\big|\; \mathbf{x} \in \mathcal{V}_j\big) + \text{P}\big(\mathbf{x} \in \{\mathcal{X}^e \setminus \mathcal{V}_j^e\} \;\big|\; \mathbf{x} \in \mathcal{V}_j\big) \tag{7.21}$$

since the second term is zero because intersection set $\{\mathcal{X}^e \setminus \mathcal{V}_j^e\}| \cap \mathcal{V}_j$ is the empty set. Hence, applying this result to equation (7.20) we obtain

$$\text{P}\big(\mathbf{x} \in \mathcal{X}^e \;\big|\; \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\big) = \sum_{\forall i} \text{P}\big(\mathbf{x} \in \mathcal{V}_i^e \;\big|\; \mathbf{x} \in \mathcal{V}_i\big)\, \text{P}\big(\mathbf{x} \in \mathcal{V}_i \;\big|\; \mathbf{x} \in \Gamma_{\text{N}}^{\text{K}}\big) \tag{7.22}$$

which is the equation in (7.11).

Appendix 7.B Theorem proof

Here we give the proof of theorem 7.1 in section 7.2. First we propose in section 7.B.1 a method to model a source $\mathcal{W}$ that generates all sequences in $\mathcal{V}_{i \leftarrow j}^e$ when $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. This allows us in

appendix 7.B.2 to employ the Lebesgue measure on $\mathcal{W} \subseteq \mathbb{R}^{\mathrm{K}}$ to measure $\mathcal{W}$ and, comparing the last measure with the measure of $\mathcal{V}_i$, draw conclusions about equation (7.14). Afterwards, in appendix 7.B.3 we extend the model to cases when the support sets of both sub-spaces overlap $|\mathcal{S}_i \cap \mathcal{S}_j| > 0$.

7.B.1 Null-space characterization of the erroneous sets

Each sequence $\mathbf{x}_i \in \mathcal{V}^e_{i \leftarrow j} \subseteq \mathcal{V}_i$ has a related $\mathbf{x}_j \in \mathcal{V}^e_{j \leftarrow i} \subseteq \mathcal{V}_j$ with a common compressed sequence $\mathbf{y} \in \mathcal{Y}$

$$\mathbf{A}\mathbf{x}_i = \mathbf{A}\mathbf{x}_j = \mathbf{y}$$

when the encoder is the typical matrix-vector product. We refer to this pair of sequences as $(\mathbf{x}_i,\mathbf{x}_j)$. The last expression can be rewritten as

$$\mathbf{A}_i\mathbf{x}_i = \mathbf{A}_j\mathbf{x}_j = \mathbf{y} \tag{7.23}$$

where the matrices $\mathbf{A}_i$ and $\mathbf{A}_j$ are $\mathrm{M} \times \mathrm{K}$ matrices generated by selecting the row vectors indicated by the support sets of $\mathcal{V}_i$ and $\mathcal{V}_j$. The vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ are composed each one by the non-null entries of the original $\mathbf{x}_i$ and $\mathbf{x}_j$.

We can rewrite equation (7.23) as

$$\mathbf{A}_i\mathbf{x}_i - \mathbf{A}_j\mathbf{x}_j = \mathbf{A}_u\mathbf{x}^*_u = 0 \tag{7.24}$$

where $\mathbf{A}_u = [\mathbf{A}_j,\ \mathbf{A}_i]$ is composed by all the column vectors of $\mathbf{A}_j$ and $\mathbf{A}_i$ and $\mathbf{x}^*_u = [\mathbf{x}_j,\ -\mathbf{x}_i]$ is the concatenation of the vectors and belongs to the null-space of the matrix $\mathbf{A}_u$, $\mathbf{x}^*_u \in \mathcal{A}^*_u \subseteq \mathbb{R}^{2\mathrm{K}}$. Note that the matrix has at most 2K column vectors when the set $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$. Now we focus on this case and later, in the sub-appendix 7.B.3 we extend the proof to the case when the supports are not disjoints.

Lemma 7.2. *If spark of measurement matrix is* $\mathrm{K}+1$ *then each* $\mathbf{x}_i \in \mathcal{V}_i$ *has at most one* $\mathbf{x}_j \in \mathcal{V}_j$ *so as equation (7.23) holds.*

Basically, $\mathbf{A}_i\mathbf{x}_i$ and $\mathbf{A}_j\mathbf{x}_j$ are unique because the rank of $\mathbf{A}_i$ and $\mathbf{A}_j$ is K, so there is a one-to-one mapping between the sets $\mathcal{V}^e_{i \leftarrow j}$ and $\mathcal{V}^e_{j \leftarrow i}$. Then it follows that

Lemma 7.3. *Each pair of sequences* $(\mathbf{x}_i,\mathbf{x}_j)$ *has a unique representation in* $\mathcal{A}^*_u$.

Thus, each sequence $\mathbf{x}_u$ can be decomposed in two unique vector vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, hence the mapping between the set of points $(\mathbf{x}_i,\mathbf{x}_j)$ is injective. Then

Lemma 7.4. *The mapping between the null space of* $\mathbf{A}_u$*,* $\mathcal{A}_u^*$*, and the set* $\mathcal{V}_{i\leftarrow j}^e$ *is injective.*

In case that spark($\mathbf{A}$) = M′ + 1[1], M ≥ M′ ≥ K it follows that we can construct a generation matrix $\mathbf{A}_u^*$ of the null-space of $\mathbf{A}_u$ with $k' = 2\mathrm{K} - \mathrm{M}'$ linear independent vectors of length 2K $\mathbf{w}$ so we can write $\mathbf{x}_u^*$ in terms of a vector of length k', $\mathbf{x}_u^* = \mathbf{A}_u^*\mathbf{w}$. We decompose matrix $\mathbf{A}_u^*$ into two sub-matrices $\mathbf{A}_{i\leftarrow j}^*$, $\mathbf{A}_{j\leftarrow i}^*$ each one with the K row vectors that generates the part of $\mathbf{x}_u^*$ that corresponds to the entries of $\mathbf{x}_i$ and $\mathbf{x}_j$ and obtain

$$\begin{bmatrix} \mathbf{x}_i \\ -\mathbf{x}_j \end{bmatrix} = \mathbf{x}_u^* = \mathbf{A}_u^*\mathbf{w} = \begin{bmatrix} \mathbf{A}_{i\leftarrow j}^* \\ \mathbf{A}_{j\leftarrow i}^* \end{bmatrix} \mathbf{w}$$

and then

$$\mathbf{x}_i = \mathbf{A}_{i\leftarrow j}^*\mathbf{w}', \;\; \mathbf{x}_j = -\mathbf{A}_{j\leftarrow i}^*\mathbf{w}' \tag{7.25}$$

Now we need to guarantee that both $\mathbf{A}_{j\leftarrow i}^*$ and $\mathbf{A}_{i\leftarrow j}^*$ have maximum rank k'. It follows that

Lemma 7.5. *The spark of matrix* $\mathbf{A}_u^*$ *is equal to* $k' + 1$ *whenever the spark of matrix* $\mathbf{A}_u$ *is larger than* K + 1*.*

Proof. By construction matrix $\mathbf{A}_u^*$ has rank k' but we need to guarantee that both $\mathbf{A}_{j\leftarrow i}^*$ and $\mathbf{A}_{i\leftarrow j}^*$ have rank k'. This happens whenever the spark of $\mathbf{A}_u^*$ is equal than $k' + 1$.

Let us assume the contrary, that the matrix has spark less than $k' + 1$. In this case there exist a sub-matrix $k' \times k'$ sub-matrix of $\mathbf{A}_u^*$ with rank less than k'. In this situation the image of a non-zero vector $\mathbf{v} \in \mathbb{R}^{k'}$ that belongs to the null space of $k' \times k'$ sub-matrix is vector zero, and hence the projection $\mathbf{v}$ onto $\mathbf{A}_u^*$ will have at least k' zeros, i.e. at most

$$\|\mathbf{A}_u^*\mathbf{v}\|_0 \leq 2\mathrm{K} - k'$$

non-null coefficients. Thus, as k' is the dimension of the null-space of $\mathbf{A}_u$,

$$k' = 2\mathrm{K} - \mathrm{M}'$$

the projection of vector $\mathbf{v}$ can have at most

$$\|\mathbf{A}_u^*\mathbf{v}\|_0 \leq 2\mathrm{K} - 2\mathrm{K} + \mathrm{M}' = \mathrm{M}'$$

1 That the spark is M′ + 1 forces that all sub-matrices of M′ vectors are linear independent, and hence the rank is of $\mathbf{A}$ is M′.

which means that the projection of $\mathbf{v}$ does not belong to the null space of $\mathbf{A}_u$ because the rank of all sub-matrices of $\mathbf{A}_u$ is M'

$$\|\mathbf{A}_u \mathbf{A}_u^* \mathbf{v}\|_0 > 0 \tag{7.26}$$

and then $\mathbf{A}_u^*$ is not a basis of the null space because it can generate vectors that do not belong to the null-space. □

From this deduction it follows that the spark$(\mathbf{A}_u^*) = k' + 1$ because otherwise the generator matrix of the null-space generates sequences that do not belong to the null-space $\mathbf{A}_u$.

From lemma 7.5 it follows that matrix $\mathbf{A}_{i\leftarrow j}^* \in \mathrm{R}^{\mathrm{K}\times k'}$ in equation (7.25) has also spark $k' + 1$, as it inherits the spark of $\mathbf{A}_u^*$. Now we are in conditions to say that $\mathbf{A}_{i\leftarrow j}^* \mathcal{W} \subseteq \mathcal{V}_{i\leftarrow j}^e$, so we can define $\mathcal{V}_{i\leftarrow j}^e$ also as follows

$$\mathcal{V}_{i\leftarrow j}^e \triangleq \left\{ \mathbf{x}_i \in \mathcal{V}_{i\leftarrow j}^e : \ \mathbf{x}_i = \mathbf{A}_{i\leftarrow j}^* \mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^{k'} \right\} \tag{7.27}$$

7.B.2 Lebesgue measures of the null-space

Now we are ready to compute the term $\mathrm{P}\left(\mathbf{x} \in \mathcal{V}_{i\leftarrow j}^e \,\middle|\, \mathbf{x} \in \mathcal{V}_i\right)$ in equation (7.14)

$$\mathrm{P}\left(\mathbf{x} \in \mathcal{V}_{i\leftarrow j}^e \,\middle|\, \mathbf{x} \in \mathcal{V}_i\right) = \frac{\mu_{\mathrm{K}}\left(\mathcal{V}_{i\leftarrow j}^e\right)}{\mu_{\mathrm{K}}\left(\mathcal{V}_i\right)}$$

where $\mu_{\mathrm{K}}(\cdot)$ is the Lebesgue measure of dimension K. If we assume that $\mathcal{V}_i \subset \mathbb{R}^{\mathrm{K}}$ is a rectangle with a side $s < +\infty$ then $\mathcal{V}_i$ is measurable with measure $0 < \mu_{\mathrm{K}}(\mathcal{V}_i) < +\infty$, as any rectangle embedded in $\mathbb{R}^{\mathrm{K}}$ is measurable.

Now let us analyze the term $\mu_{\mathrm{K}}\left(\mathcal{V}_{i\leftarrow j}^e\right)$ as a function of the number of samples M and the dimension of the $\mathbb{R}^{\mathrm{K}}$ sub-spaces $\mathcal{V}_i$ embedded into the Grassmannian manifold $\Gamma_{\mathrm{N}}^{\mathrm{K}}$. As stated before $\mathbf{A}_{i\leftarrow j}^* \mathcal{W} \subseteq \mathcal{V}_{i\leftarrow j}^e$, so it follows that

$$\mu_{\mathrm{K}}\left(\mathcal{V}_{i\leftarrow j}^e\right) \geq \mu_{\mathrm{K}}\left(\mathbf{A}_{i\leftarrow j}^* \mathcal{W}\right)$$

and the equality follows when $\mathcal{W} \subseteq \mathbb{R}^{k'}$.

Now we need to deduce the value of $\mu_{\mathrm{K}}\left(\mathcal{V}_{i\leftarrow j}^e\right)$ in different cases: $2\mathrm{K} \leq \mathrm{M}'$, $\mathrm{K} = \mathrm{M}'$ and $\mathrm{K} < \mathrm{M}' < 2\mathrm{K}$.

Let us recall that $k' = 2\mathrm{K} - \mathrm{M}'$, $\mathrm{M}' \geq \mathrm{K}$ and that $\mathbf{A}_u$ has dimension $\mathrm{M} \times 2\mathrm{K}$ and that $\mathrm{spark(A)} = \mathrm{M}' + 1$ from appendix 7.B.1.

Case: $2\mathrm{K} \leq \mathrm{M}'$

In this case $\mathbf{A}_u$ is a $\mathrm{M} \times 2\mathrm{K}$, with rank 2K so the null-space contains only the zero vector ($\mathcal{V}_i^e$ only contains the zero vector) and so the mapping is injective. Note that this result coincides with the one in lemma (3.1) when $\mathrm{M}' = \mathrm{M}$.

Case: $\mathrm{K} = \mathrm{M}'$

In this case $k' = \mathrm{K}$ because $k' = 2\mathrm{K} - \mathrm{M}'$. Let us analyze $\mu\left(\mathcal{V}^e_{i\leftarrow j}\right)$ when that dimension of the null-space of $\mathbf{A}_u$ is larger than the one of $\mathcal{V}_i$, i.e. the trivial case of $\mathrm{K} \geq \mathrm{M}'$. In this case $\mathbf{A}^*_{i\leftarrow j}$ is a matrix with K rows and k' columns with spark $\mathrm{M}' + 1$, i.e. the rank of all $\mathrm{K} \times \mathrm{K}$ sub-matrices is $\mathrm{K} = \mathrm{M}'$.

Assume that $\mathbf{A}_{i\leftarrow j} = \mathbf{U}\Sigma\mathbf{V}^t$ is the SVD decomposition of $\mathbf{A}_{i\leftarrow j}$. Hence, $\mathbf{U}$ and $\mathbf{V}$ are a K and a k' orthogonal square matrices. Then

$$\mu_{\mathrm{K}}\left(\mathbf{A}^*_{i\leftarrow j}\mathcal{W}\right) = \mu_{\mathrm{K}}\left(\mathbf{U}\Sigma\mathbf{V}^t\mathcal{W}\right) = \mu_{\mathrm{K}}\left(\Sigma\mathcal{W}\right) \tag{7.28}$$

where we applied that the Lebesgue measure of a set is preserved by the projection onto an orthogonal matrix. Therefore, whenever the set $\mathcal{W}$ is measurable with the Lebesgue measure of dimension k' and has measure $\mu_{k'}(\mathcal{W}) > 0$ the set $\Sigma\mathcal{W}$ in equation (7.28) spans all $\mathbb{R}^{\mathrm{K}}$, so all sequences $\mathbf{x} \in \mathcal{V}_i$ will also be in $\mathcal{V}^e_{i\leftarrow j}$. It follows that

$$\mathrm{P}\left(\mathbf{x} \in \mathcal{V}^e_{i\leftarrow j} \mid \mathbf{x} \in \mathcal{V}_i\right) = \frac{\mu_{\mathrm{K}}\left(\mathcal{V}^e_{i\leftarrow j}\right)}{\mu_{\mathrm{K}}\left(\mathcal{V}_i\right)} = \frac{\mu_{\mathrm{K}}\left(\mathcal{V}_i\right)}{\mu_{\mathrm{K}}\left(\mathcal{V}_i\right)} = 1 \tag{7.29}$$

therefore, the mapping cannot be injective.

Case: $\mathrm{K} < \mathrm{M}' < 2\mathrm{K}$

Let us analyze now the measure $\mu_{\mathrm{K}}\left(\mathbf{A}^*_{i\leftarrow j}\mathcal{W}\right)$ for the remaining case, when $\mathrm{K} > \mathrm{k}' > 0$. It can be shown that $\mu_{\mathrm{K}}\left(\mathbf{A}^*_{i\leftarrow j}\mathcal{W}\right) = 0$ because $\mathbf{A}^*_{i\leftarrow j}$ is a tall matrix. In this case we can add $\mathrm{K} - k'$ columns with zeros, obtaining a matrix $\mathbf{P}$, and assume that is mapped somehow into $\mathbb{R}^{\mathrm{K}}$ preserving the original volume so as for all $\mathbf{v}$ in $\mathcal{W}$ and its extended version $\mathbf{v}'$, $\mathbf{A}^*_{i\leftarrow j}\mathbf{v} = \mathbf{P}\mathbf{v}'$. Then

$$\begin{aligned}\mu_{\mathrm{K}}\left(\mathbf{A}^*_{i\leftarrow j}\mathcal{W}\right) &= \mu_{\mathrm{K}}\left(\mathbf{P}\mathcal{W}\right) \\ &= \det\left(\mathbf{P}\right)\mu_{\mathrm{K}}\left(\mathcal{W}\right) = 0\end{aligned} \tag{7.30}$$

because $\det(\mathbf{P}) = 0$. Hence

$$\mathrm{P}\left(\mathbf{x} \in \mathcal{V}^e_{i\leftarrow j} \;\middle|\; \mathbf{x} \in \mathcal{V}_i\right) = \frac{\mu_{\mathrm{K}}\left(\mathcal{V}^e_{i\leftarrow j}\right)}{\mu_{\mathrm{K}}\left(\mathcal{V}_i\right)} = 0$$

so $\mathrm{P}\left(\mathbf{x} \in \mathcal{X}^e \;\middle|\; \mathbf{x} \in \Gamma^{\mathrm{K}}_{\mathrm{N}}\right) = 0$ in equation (7.14), i.e. we say that the event of drawing a sequence $\mathbf{x}$ of the Grassmanian manifold $\Gamma^{\mathrm{K}}_{\mathrm{N}}$ which is in the set of non-injective sequences $\mathbf{x} \in \mathcal{X}^e$ happens with probability zero, so in this case the perfect reconstruction of all the sequences cannot be achieved, as *only* can be compressed without losses *almost* all the sequences in $\Gamma^{\mathrm{K}}_{\mathrm{N}}$.

Note that equation 7.30 holds whenever $2\mathrm{K} \geq \mathrm{spark}(\mathbf{A}) > \mathrm{K}+1$ because the $\mathrm{spark}(\mathrm{A}) = \mathrm{M}'+1$.

7.B.3 Extension to non-disjoint partitions

Now, let us check what happens when $\mathcal{V}_i \cap \mathcal{V}_j \neq \emptyset$. On the one hand, the dimension of the null-space of $\mathbf{A}_u$ is lower than before because $k'' = 2\mathrm{K} - \mathrm{M}' - |\mathcal{S}_i \cap \mathcal{S}_j|$ i.e. $k'' = k' - |\mathcal{S}_i \cap \mathcal{S}_j|$. On the other hand, as the $\mathcal{S}_i \cap \mathcal{S}_j$ is not empty, the decomposition of the sequences $\mathbf{x}^* \in \mathcal{A}^*_u$ in two vectors with disjoint supports cannot be done anymore, i.e. $\mathbf{x}^* \neq [\mathbf{x}_i,\ \mathbf{x}_j]$. $\mathbf{x}^* = \mathbf{x}_i + \mathbf{x}_j$ is still a vector of the null-space but now we have to decompose the vector in three parts $\mathbf{x}^* = \left[\mathbf{x}'_i,\ \mathbf{x}_{i,j},\ \mathbf{x}'_j\right]$ and take into account that $\mathbf{x}_i = [\mathbf{x}'_i,\ \mathbf{x}''_i]$ and $\mathbf{x}_j = \left[\mathbf{x}''_j,\ \mathbf{x}'_j\right]$ so the $|\mathcal{S}_i \cap \mathcal{S}_j|$ coefficients of $\mathbf{x}^*$ in the intersection set must accomplish that $\mathbf{x}_{i,j} = \mathbf{x}''_i + \mathbf{x}''_j$. Hence, the coefficient $\mathbf{x}''_i$ can take any value among all the sequences of $\mathbb{R}^{|\mathcal{S}_i \cap \mathcal{S}_j|}$, so now for each sequence $\mathbf{x}^* \in \mathcal{A}^*_u$, i.e $\mathbb{R}^{|\mathcal{S}_i \cap \mathcal{S}_j|} \subset \mathcal{V}^e_{i\leftarrow j}$.

The latter can be also modeled with a matrix. Equation (7.31) shows how matrix $\mathbf{A}^*_{i\leftarrow j}$ can be constructed according to the model

$$\begin{bmatrix} \mathbf{P}; & \begin{matrix}\mathbf{0}\\ -\mathbf{I}\end{matrix} \end{bmatrix} = \mathbf{A}^*_{i\leftarrow j} \tag{7.31}$$

where $\mathbf{P}$ is a $\mathrm{K} \times k''$ matrix with the rows vectors that generate $\mathcal{V}_i$ and matrix $\mathbf{I}$ is an identity matrix with $|\mathcal{S}_i \cap \mathcal{S}_j|$ rows and columns. Thus, matrix $\mathbf{A}^*_{i\leftarrow j}$ is again a $\mathrm{K} \times \mathrm{k}'$ matrix with rank k'.

Conversely, matrix $\mathbf{A}^*_{j\leftarrow i}$ is constructed as shown below

$$\begin{bmatrix} \begin{matrix}\mathbf{0}\\ \mathbf{P}'\end{matrix} ; & \begin{matrix}\mathbf{I}\\ \mathbf{0}\end{matrix} \end{bmatrix} = \mathbf{A}^*_{j\leftarrow i} \tag{7.32}$$

where $\mathbf{P}'$ is a matrix with $\mathrm{K} - |\mathcal{S}_i \cap \mathcal{S}_j|$ that belong to $\mathcal{V}_j$, but the one that are already in $\mathbf{P}$ of equation (7.32), and k'' columns and matrix $\mathbf{I}$ is an identity matrix with $|\mathcal{S}_i \cap \mathcal{S}_j|$ rows and columns, as before. Note that matrix $\mathbf{A}^*_{j\leftarrow i}$ has also rank k'.

We can generate the pair $(\mathbf{x}_j,\mathbf{x}_j)$ for each $\mathbf{x}_j \in \mathcal{V}^e_{j\leftarrow i}$ and $\mathbf{x}_i \in \mathcal{V}^e_{i\leftarrow j}$ as

$$\mathbf{x}_j = \mathbf{A}^*_{j\leftarrow i}\mathbf{v}' \qquad \mathbf{x}_i = \mathbf{A}^*_{i\leftarrow j}\mathbf{v}' \tag{7.33}$$

where $\mathbf{v}' = \left[\mathbf{v}, \mathbf{x}''_j\right] \in \mathbb{R}^{k'}$.

Let us recall gain the assumptions we made in appendix 7.B.1: $k' = 2\mathrm{K} - \mathrm{M}'$, $\mathrm{M}' \geq \mathrm{K}$ and that $\mathbf{A}_u$ has dimension $\mathrm{M} \times 2\mathrm{K}$ and that $\mathrm{spark(A)} = \mathrm{M}' + 1$. Then, as shown in equation (7.30), any set $\mathcal{V}^e_{i\leftarrow j}$ has K dimensional Lebesgue measure zero because $\mathrm{K} > \mathrm{k}'$ since the $\mathrm{spark}(\mathbf{A}) = \mathrm{M}' + 1$, as we showed at the end of appendix 7.B. Hence,

$$\mathrm{P}\left(\mathbf{x} \in \mathcal{V}^e_{i\leftarrow j} \;\middle|\; \mathbf{x} \in \mathcal{V}_i\right) = \frac{\mu_{\mathrm{K}}\left(\mathcal{V}^e_{i\leftarrow j}\right)}{\mu_{\mathrm{K}}\left(\mathcal{V}_i\right)} = 0$$

and so, almost lossless compression can be also achieved in this case as $\mathrm{P}\left(\mathbf{x} \in \mathcal{X}^e \;\middle|\; \mathbf{x} \in \Gamma^{\mathrm{K}}_{\mathrm{N}}\right) = 0$ in equation (7.14).

CHAPTER 8

Verification-based algorithms

In this chapter several contributions are done to the field of compression schemes for noiseless compressed sensing: we present here two verification-based algorithms and a sparse structured matrix specially design for sequential sampling schemes for noiseless compressed sensing.

This chapter is organized as follows: in section 8.1 we review the verification algorithm (VA), proposed in [166]. Then, we present two novel verification-based algorithms for noiseless compressed sensing. First, in section 8.2 we present a variable-length compression scheme for noiseless compressed sensing. We introduce a novel approach to the problem: a measurement matrix is specially designed to sense equally early all the components of the original vector $\mathbf{x}_0$ and the enhanced verification algorithm (EVA), a local message passing strategy aimed at mitigating the measurement matrix sparsity loss required to deal with very sparse signals. Second, we present the list message passing for compressed sensing (CS-LMP) algorithm in section 8.3. Up to the knowledge of the author, this is the best performing verification-based algorithm in the literature. We also enclose in appendices 8.A.2 and 8.A.3 and extended explanation of the implementation of the algorithm of the EVA and the CS-LMP algorithms, respectively.

8.1 Introduction

8.1.1 System setup

In this chapter we assume that the signal vector $\mathbf{x}_0$ is drawn at random from the Grassmanian manifold Γ_N^K, $0 \leq K < N$. The coefficients indexed in $\mathcal{S}$ are randomly drawn according to a probability density function. Throughout all this chapter we assume that the encoder is linear and measures are not corrupted by noise $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$.

The algorithms reviewed and presented here are message passing strategies over the graph representation of the measurement matrix $\mathbf{A}$. Unless explicitly stated, $\mathbf{A}$ is chosen randomly from an ensemble Γ(L,R,N). The graph representation of $\mathbf{A}$ is free of length-four cycles whenever it is possible. The performance of the verification-based algorithms do not depend of the values of the non-null coefficients of matrix $\mathbf{A}$ whenever it is free of length-four cycles.

8.1.2 Verification algorithm for compressed sensing

Here we review the verification algorithm (VA) in [166]. This algorithm is aimed at reconstructing a sparse signal vector $\mathbf{x}_0$ in the set-up of section 8.1.1 and has computational complexity O(N). This algorithm was identified by [166] to be a different implementation of the sudocode decoding algorithm [132]. On one side, it has a lower computational complexity in certain cases, $O(K^2)$. On the other side, VA implementation only requires local operations, as opposed to the sudocodes [132].

When applied in the CS framework, VA exploits that

(a) There is no noise.

(b) The probability density function of the components of vector $\mathbf{x}_0$ is a mixture of a discrete and continuous probability density function with a unique mass concentration at zero $P(X = 0) = 1 - K/N = 1 - \gamma$.

(c) The graph has no length-four cycles.

Due to these facts the following statements hold:

S1 If a measurement y is zero, all the variable nodes indexed by $\mathcal{J}^c$ are zero.

S2 When the graph does not exhibit cycles of length four and two measurements y_j and y_n are identical ($y_j = y_n$), that pair of measurements share a single variable node, $|\mathcal{J}_j^c \cap \mathcal{J}_n^c| = 1$. Therefore, it follows that the common component in the intersection set must be equal to y_j and the remaining variable nodes in $\mathcal{J}_j^c$ and $\mathcal{J}_n^c$ must be equal to zero.

Keeping these statements in mind, the algorithm works as follows. Check (variable) nodes exchange with variable (check) nodes messages having the following form: {*state,value*} where *state*='v' indicates that we know for sure that the value of the associated variable node is *value* (i.e. the variable node is *verified*) whereas *state*='nv' informs that *value* is solely an estimate of the variable node value (i.e. the variable node is still *not verified*).

A check node is said to *verify* a neighboring variable node whenever the first can infer with infinite reliability the value of the latter using statements S1 and S2. Then, the verified variable

node propagates its state to the other check nodes in its neighborhood. Hence, these check nodes can remove the contribution of the verified variable node from their respective measurement and try to infer, in the next round, the values of the remaining variable nodes connected to them.

The iterative process starts with variable nodes sending a *non-verified* message with dummy value $-$,{'nv',$-$}, to all the check nodes connected to them. Then, check nodes are activated and send a message to variable nodes in their neighborhood. Here we briefly describe the verification algorithm, whereas in appendix 8.A.1 we enclose an extended description of how messages are updated by check and variable nodes in algorithms 8.1 and 8.2, respectively.

First of all, each check node removes the contributions of the incoming *verified* messages from its measurement y to obtain y' and then, updates all the edges connected to variable nodes that remain not verified (indexed in the set $\mathcal{J}_{nv}^{c}$ [1]), according to rules C1, C2 and C3:

C1: If $y' = 0$, propagate to all the variable nodes indexed by $\mathcal{J}_{nv}^{c}$: {'v', 0}.

C2: If $|\mathcal{J}_{nv}^{c}| = 1$, propagate to the variable nodes in $\mathcal{J}_{nv}^{c}$: {'v', y'}.

C3: Otherwise, propagate to all variable nodes indexed by $\mathcal{J}_{nv}^{c}$: {'nv', y'}.

Rule C1 exploits statement S1. Rule C2 deals with the trivial case in which there is a single non-verified variable node connected to the check node. Rule C3 deals with the remaining cases: whenever the check node cannot infer the value of a variable node it propagates the *non-verified* message with the updated measurement, ('nv',y').

After check nodes update, variable nodes are activated and proceed as follows:

V1: If the variable node receives a *verified* message, then propagate this message through the remaining edges {'v', 'value'}.

V2: If the variable node receives at least two *non-verified* messages with the same estimate 'value', then propagate {'v', 'value'} to all the nodes connected to them.

V3: Else, propagate {'nv', -} through its edges.

where rule **V1** deals with verified variable nodes. Basically, due to the absence of noise, a verified variable node informs to check nodes in its neighborhood that it is verified by propagating a *verified* message along with its value. Rule **V2** exploits statement **S2**. In this case, the variable node receives more than one *non-verified* message with equal estimate y' so it can infer its value. Hence, the variable node switches its state to verified and informs of this update to all check nodes in its neighborhood by sending them the message {'v',value}. Finally, in case of rule **V3**,

1 We refer to the variable nodes in the neighborhood of a check node that remain non-verified as $\mathcal{J}_{nv}^{c}$.

the variable node just sends a *non-verified* message {'nv',−} to all the neighboring check nodes when it cannot infer its value.

8.2 Enhanced verification algorithm and sequential sampling for compressed sensing

In this section we present the variable rate graph-based CS scheme that we proposed in [117]. In this work we presented the enhanced verification algorithm (EVA), a decoding algorithm for noiseless compressed sensing along with a structured matrix specially designed for variable rate schemes where the check node degree distribution was obtained from a closed-form expression[1]. We showed that this novel approach outperforms the l_1-based theoretical limits. The proposed scheme stemmed from the original design of sudocodes [132], which has the distinctive feature of taking infinite reliability decisions (i.e. no errors are made). The main features of our compression scheme are

(a) The introduction of a block-wise constructed measurement matrix structure for the encoder.

(b) The proposal of a check node degree profile design criterion.

(c) An extension of the verification algorithm.

This section is structured as follows. First in subsection 8.2.1 we briefly introduce the proposed scheme. After that in subsection 8.2.2 we propose a systematic method to construct binary matrices for sequential sampling schemes. In the next subsection we introduce some enhancements to the VA aimed at mitigating the effect of cycles of length 4 in dense binary graphs for the noiseless setting. Finally, in subsection 8.2.5 we enclose simulation results to illustrate the performance enhancement obtained by the proposed scheme.

8.2.1 System setup

The measurement matrix is generated according to the procedure described next in section 8.2.2 and it is characterized by its check and variable degree profiles R(d) and L(d), respectively. This structured matrix tackles a problem that random matrices have in sequential schemes, since it is aimed to sense the components of vector $\mathbf{x}_0$ equally early. The compression scheme introduced here uses the same matrix $\mathbf{A}$ for all signal realizations. We restrict $\mathbf{A}$ to be a binary matrix

1 The check node degree distributions are chosen in a similar manner to the policy of the group testing scheme presented in chapter 5.

where the number of ones increases as the sparsity ratio γ decreases and so it becomes denser for signals with very low sparsity ratio.

The binary and sparsity features of $\mathbf{A}$ allow retrieving the signal vector $\mathbf{x}_0$ with a low-complexity decoding algorithm based on the description of the system by a graph. The decoding attempts are done using the verification-based message passing algorithm introduced in subsection 8.2.4. We also enclose an extended description of this algorithm in appendix 8.A.2.

The proposed system employs a sequential non-adaptive sampling procedure, i.e., measurements are drawn one by one following the order of the rows of matrix $\mathbf{A}$ and after each new measurement is drawn a new attempt is made to recover the sequence $\mathbf{x}_0$. This procedure is followed until all the components in $\mathbf{x}_0$ are identified. Note that the number of measurements required to reconstruct vector $\mathbf{x}_0$ may be different for each signal realization.

This enhancement of the decoding algorithm in [166] is particularly relevant for very sparse signals, since in this case length-four cycles cannot be avoided in the graph. The proposed algorithm outperforms the results in [132] and [166] in terms of number of measurements while having similar complexity, and has lower complexity than [135].

8.2.2 Structured matrix for sequential sampling

We design the distribution $\mathrm{L}(d)$ to be regular, i.e. all the variable nodes are connected to the same number d_c of check nodes, whereas the distribution $\mathrm{R}(d)$ is optimized in section 8.2.3. The resulting check node degree profile $\mathrm{R}(d)$ has at most two consecutive degrees.

Rather than generating matrix $\mathbf{A}$ as a random realization of the ensemble of matrices having degrees profiles $\mathrm{R}(d)$ and $\mathrm{L}(d)$, we chose to introduce some structure in the construction of matrix $\mathbf{A}$ to guarantee that all components of vector $\mathbf{x}_0$ contribute to a measurement before any of them is sensed again. This structure is beneficial in the sequential sampling procedure: it ensures that all the components of vector $\mathbf{x}_0$ are sensed equally early, avoiding the possibility that a non-zero sample may not be sensed until late and, therefore, the sequential sampling may require a large number of samples to stop.

Hence, we propose to construct a matrix $\mathbf{A}$ by sub-blocks. Let us assume, for the sake of simplicity that all rows have the same degree d_c. Then, $d_v = \mathrm{M}d_c/\mathrm{N}$ sub-matrices $\mathbf{A}_1 \dots \mathbf{A}_{d_v}$ of dimension $\mathrm{N}/d_v \times \mathrm{N}$ are randomly drawn from the ensemble of all matrices of row degree d_c and column degree 1. Afterwards, these sub-matrices are stacked to build matrix $\mathbf{A}$ as $\mathbf{A} = [\mathbf{A}_1^{\mathrm{T}} \dots \mathbf{A}_{d_v}^{\mathrm{T}}]^{\mathrm{T}}$.

The design of sub-matrices $\mathbf{A}_1 ... \mathbf{A}_{d_v}$ is done avoiding length-four cycles whenever possible. Unfortunately, when the signal is very sparse these cycles are unavoidable if one wants to keep the number of measurements low, since the vector length N is finite. Indeed, if the number

of measurements is in the order of K the check node degree d_c must be O(N/K) in order to guarantee that all signal coefficients are sensed once, so the check node degree d_c grows as the signal becomes sparse and hence matrix $\mathbf{A}$ becomes denser and length-four cycles become unavoidable for finite length N. This behavior explains the relevance of the enhancement of the decoding algorithm proposed next in section 8.2.4.

8.2.3 Check node degree profile design

As stated in section 2.1.2, the noiseless compression schemes and the group testing schemes are strongly related because the group testing problem can be seen as a 1 bit quantization of the compressed sensing problem when the matrix is sparse. Therefore, when the matrices are sparse the performance of both compression schemes depends critically on the amount of non-null components/defective subjects that contribute to each measure/test.

Let us recall equations (2.4,2.5, 2.6) in section 2.3.3 that they describe the probabilities $\mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| = 0\big)$, $\mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| = 1\big)$ and $\mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| > 1\big)$. We have simply to note that the probability of non-defective subjects in section 2.3.3 translates here to $\mathrm{P}(\mathcal{P}_N) = 1 - \gamma$.

For group testing we employed a policy aimed to maximize the entropy vector $\mathbf{y}_0$ whereas here we decided to select a check node degree profile $\mathrm{R}(d)$ aimed at balancing the amount of vectors generated according to equations (2.4) and (2.5).

With this purpose, we propose to employ the check node degree distribution $\mathrm{R}(d)$ that maximizes the average number of variable nodes connected to check nodes for which $|\mathcal{J}^c \cap \mathcal{S}| \leq 1$, as follows

$$\begin{aligned} \underset{\mathrm{R}(d)}{\text{maximize}} \quad & \sum_{\forall d} \mathrm{R}(d)\, d\, \mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| \leq 1\big) \\ \text{subject to} \quad & \mathrm{R}(d) \geq 0, \\ & \sum_{\forall d} \mathrm{R}(d) = 1 \end{aligned} \tag{8.1}$$

where $\mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| \leq 1\big) = \mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| = 0\big) + \mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| = 1\big)$. Note that $\mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| \leq 1\big)$ is the fraction of check nodes with $|\mathcal{J}^c \cap \mathcal{S}| \leq 1$ for asymptotic N. Then if follows that $d\, \mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| \leq 1\big)$ is the average number of edges per check node so as $|\mathcal{J}^c \cap \mathcal{S}| \leq 1$ for a given $1 - \gamma$ and d.

It is straightforward to show that for a fixed $1-\gamma$ and $d \in \mathbb{R}^+$ the relaxed function $d\, \mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| \leq 1\big)$ has a single maximum at $\bar{d}$. It can be shown that whenever we fix the number of non-null coefficients of $\mathrm{R}(d)$ to d and $d+i$ and assume that $\mathrm{N/K} >> 1$, the optimum choice of the two non-null degrees are the closest ones around $\bar{d}$. In this case $\sum \mathrm{R}(d) d\, \mathrm{P}\big(|\mathcal{J}^c \cap \mathcal{S}| \leq 1\big)$ has a unique solution $\mathrm{R}(d)$ that can be inferred from $\bar{d}$ as follows

$$\mathrm{R}(\lfloor \bar{d} \rfloor) = \lceil \bar{d} \rceil - \bar{d}, \;\; \mathrm{R}(\lceil \bar{d} \rceil) = \bar{d} - \lfloor \bar{d} \rfloor \tag{8.2}$$

$\bar{d}$ can be calculated as

$$\begin{aligned} \bar{d} &= -t_1(1+t_2)/2 \\ t_1 = \frac{1-\gamma}{\gamma} + \frac{2}{\log(1-\gamma)}, &\qquad t_2 = \sqrt{1 - \frac{4(1-\gamma)}{\gamma \log(1-\gamma) t_1^2}} \end{aligned} \tag{8.3}$$

Note that (8.3) only requires the knowledge of the sparsity ratio of $\mathbf{x}_0$. It is worth mentioning that $\bar{d}$ also increases as N/K increases.

The idea behind equation (8.1) is that by verifying as many variable nodes per measurement as possible, the phase transition zone[1] will be reached as soon as possible (in terms of samples). The EVA algorithm shows a kind of avalanche effect when it reaches its phase transition zone: once it reaches this zone the addition of a single sample may enable the verification of a large amount of variable nodes. This happens because rule C2 in algorithm 8.A.2 is activated when the average number of non-verified variable nodes connected to check nodes is close to 1. When this average is close to one, the avalanche effect is triggered. Before that, variable nodes can only be verified in the graph when statements S1 and S2 in section 8.1.2 our modified S2 in section 8.2.4 happen. Hence, by maximizing the fraction of variable nodes connected to check nodes with $|\mathcal{J}^c \cap \mathcal{S}| \leq 1$ we try to reduce as fast as possible the number of non-verified variable nodes while keeping low the number of generated measurements.

8.2.4 Enhanced verification decoder

Now let us focus on the modification of the verification algorithm that we proposed in [117]. There we showed that this modification improves performance of the VA when dealing with very sparse sources while preserving its O(N) computational complexity and requiring only local operations.

More specifically, the improvement consists in extending rule V2 in subsection 8.1.2 to deal with graphs having cycles of length four, i.e., $|\mathcal{J}_j^c \cap \mathcal{J}_i^c| \geq 1$. In the presence of such cycles, if two measurements y_j and y_i take the same value ($y_j = y_i$), we know that the variable nodes that are not in the intersection set $\mathcal{J}_j^c \cap \mathcal{J}_i^c$ are equal to zero but we cannot determine the individual value of those variable nodes in the intersection set; in fact, we can only affirm that the variable nodes that are in the intersection set must sum up to y_j. In this case, it is important that the variable

1 See section 3.2.1 for further details.

nodes in the intersection inform the check nodes of this *coincidence*, so that the variable nodes that do not belong to the intersection set can be verified as zero.

To implement the above mechanism described in 8.A.2, we have modified the original message passing algorithm including a new message named *coincidence* 'c'. This message is generated by the variable nodes that detect coincident measurements and is sent to the check nodes detected to be in the *coincidence* state. Thanks to this, check nodes know that

(i) there is at least another check node with a measurement equal to its measurement and the cardinality of the intersection (the number of received coincident messages)

(ii) one or more of the nodes in coincident state sum up to the measurement.

In other words, if a check node receives only a *coincidence* message it means that there is no cycle of length four so it can verify the variable node that sent the *coincidence* message with the value of the measurement and the remaining nodes with zero (applying the original rule V2). Otherwise, if the check node receives more than one *coincidence* message, it sends a *non-verified* message to the variable nodes that sent *coincidence* messages and propagates verification messages $\{$'v',$value = 0\}$ to the remaining ones. Briefly, these changes are summarized as follows:

- *New variable node rule* **V2**: If a variable node receives at least two *non-verified* messages with the same estimate, it sends a *coincidence* message to these check nodes, and transmits $\{$'nv',$-\}$ to the remaining nodes in its neighborhood.
- *Check node rule* **C4**: If a check node receives at least a *coincidence* message, it verifies as zero the variable nodes that have not sent *coincidence* messages. Additionally, if the check node receives only a *coincidence* message, it verifies the corresponding variable node using its own measurement. Otherwise, it propagates $\{$'nv',$y'\}$ to the variable nodes that sent *coincidence* messages.

8.2.5 Performance

Table 8.1 compares the joint performance of the proposed matrix construction method, the proposed check node degree design and the proposed message passing algorithm with the performance of the sudocodes [132]. In this table, M represents the maximum number of samples required for perfect reconstruction obtained after 10^5 Monte Carlo simulations. The signal vector $\mathbf{x}_0$ was drawn at random from the Grassmanian manifold $\Gamma_{\mathrm{N}}^{\mathrm{K}}$, $0 \leq \mathrm{K} < \mathrm{N}$. The coefficients indexed in $\mathcal{S}$ are randomly drawn according to a Gaussian probability density function with zero mean and $\sigma_x^2 = 1$.

Table 8.1: Comparison of sudocodes with EVA.

N,K	Sudocodes [132]	Proposed method
10000,100	M=803	M=375
10000,10	M=461	M=85
100000,10	M=931	M=92

Note that the number of samples required by the proposed scheme is 2 to 10 times smaller that the one required by sudocodes. It must be noted that the proposed method also outperforms that one in [166] (see figure 6 in [166] where the adopted LM2-MB algorithm is the VA algorithm presented in section 8.1.2). In this case for N = 10000 the method in [166] requires M > 500 measurements to reconstruct a sequence with K = 100; in our case, M = 375 suffices to reconstruct almost any sequence with K = 100 (see table 8.1).

Figure 8.1 shows a phase transition diagram as a function of the number of samples M, sparsity K and block length N. The horizontal axis corresponds to the sampling ratio, $r = \mathrm{M/N}$. The vertical axis represents the ratio $\delta = \mathrm{K/M}$. The curve labeled as LP-PT is phase transition for l_1 reconstruction [49, 50]. The curves labeled as (VA) and (EVA) represent the performance of the verification and enhanced verification algorithms for different $p_e = \mathrm{P}\left(\mathbf{x}_0 \neq \mathbf{x}\right)$ when the proposed matrix construction is employed. The same measurement matrix was employed to run all the simulations (VA and EVA) for a fixed N/K ratio. These matrices were obtained with the structure presented in section 8.2.2 and with R(d) selected for $p = 1 - \mathrm{K/N}$ as indicated by

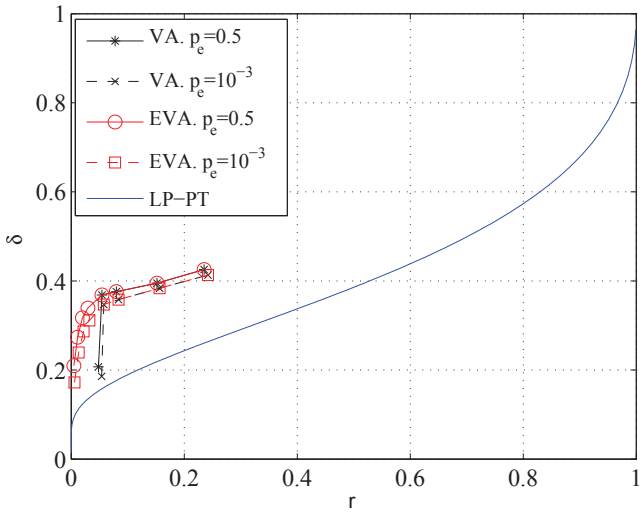


Figure 8.1: Phase transition diagram as a function of sampling, sparsity and block length N. Comparison of verification (VA) and enhanced verification (EVA) algorithm for N=16000 and several sparsity ratios on a phase diagram.

equations (8.3) and (8.2) in appendix 8.2.3. At least 10^5 Monte Carlo simulations were run per point.

The plots obtained with VA and EVA show a dual behavior: both have the same performance as long as it is possible to generate graphs without length-four cycles (for $r < 0.05$ in this case, i.e. N/K > 100). Once the measurement matrix becomes dense, i.e. for large N/K ratios, the behavior of the algorithms changes: VA performance decreases dramatically whereas EVA performance falls slowly. These results can be compared with the l_1 theoretical limit for asymptotic N. Figure 8.1 shows that VA can outperform the l_1 in an approximate range between $r \in (0.05, 0.25)$ and EVA between $r \in$ (0,0.25), both employing measurement matrices constructed with the method proposed in this chapter.

8.3 List message passing for compressed sensing

Recently, the authors of [167] proposed a new verification based-algorithm in the context of q-ary channel coding, named list message passing (LMP) . The main differences of LMP with respect to verification-based algorithms are that:

(a) Whenever the field 'state' of a message is 'non-verified' the field value is a *list of estimates* of the value of the variable node

(b) All the nodes exchange extrinsic information.

The drawback of this channel coding algorithm is that the size of the list of estimates grows unbounded with the iteration number and the check node degree.

Clearly, the complexity of direct application of LMP to CS is non-affordable because all nodes are originally unknown and because the average check node degree is inversely proportional to the source sparsity γ^{-1}. With this in mind, in this section, we present an adaptation of the list message passing algorithm for compressed sensing (CS-LMP) that overcomes this problem and aims at approaching the BP-threshold γ^{BP} (see section 3.3.2 for further details) with affordable computational complexity.

This section is divided as follows: in section 8.3.1 we briefly describe the compression scheme. Then in section 8.3.2 we introduce the CS-LMP algorithm. After that in section 8.3.3 we study its computational complexity. Finally, in section 8.3.4 we compare its performance with the NB-VA and the GAMP reconstruction algorithms.

8.3.1 System setup

Here we assume that the signal vector $\mathbf{x}_0$ is drawn at random from the Grassmanian manifold Γ_N^K, $0 < K \leq N$, i.e. all the realizations have exactly K non-null components so the components that are not indexed in $\mathcal{S}$ are strictly zero. The support set $\mathcal{S}$ is selected randomly from the $\binom{N}{K}$ possible choices according to a uniform distribution. After that, the components indexed in $\mathcal{S}$ are populated by drawing $|\mathcal{S}|$ i.i.d sampled from a continuous distribution.

Throughout all this section we assume the typical noiseless compressed sensing setup: $\mathbf{y}_0 = \mathbf{A}\mathbf{x}_0$. In this compression scheme $\mathbf{A}$ is a sparse random matrix from an ensemble Γ(L,R,N). Note that the compression scheme presented here is a fixed-to-fixed rate code where all components of vector $\mathbf{y}_0$ are generated in parallel, as opposite to the EVA presented 8.2.

The decoder implements the List-Message-Passing algorithm for CS, that we recently presented in [118], is a local message passing strategy between the nodes of the graphical representation of a sparse matrix and has the best known verification-based algorithm and the one with the highest computational complexity.

8.3.2 List message passing based-decoder for noiseless compressed sensing

In this message passing strategy nodes exchange messages having the following form: {*state,value*} where *state* is either *non-verified* 'nv' or *verified* 'v', and *value* is either a number (when 'v') or a list of its estimates of length $d_v - 1$ (when 'nv'). Check and variable nodes are activated alternatively performing iterations. The iterative process ends once all variable nodes are verified or when a stop condition is reached.

Variable nodes, when activated, execute the following tasks

V1 If a verified message is received then propagate this information to the remaining check nodes in their neighborhood, swap its state to "verified" and store the verified "value".

V2 Compare the values of the received messages. In case of receiving two or more messages with the same value swap its state to "verified" and store the common value. Finally, convey this information to all the check nodes in its neighborhood.

V3 Otherwise, forward to all check nodes in its neighborhood the 'nv' flag and the list of estimates received from the other check nodes.

During their activation turn, check nodes divide their edges into the sets of verified and non-verified edges, $\mathcal{I}_v^c$ and $\mathcal{I}_{nv}^c$, and subtract the contribution of the verified variable nodes from their measurement to obtain y' as follows

$$y'_j = y_j - \sum_{\forall i \in \mathcal{I}_{v,j}^c} a_{j,i}\, x_i \tag{8.4}$$

where y_j is the measurement of the j-th check node, $\mathcal{I}_{v,j}^c$ represents the verified variable nodes in the neighborhood of the j-th check node, $a_{j,i}$ is the weight of the edge connecting the variable node i and the check node j and x_i is the verified value of the variable node i.

After that, each check node performs a search in order to check whether any combination of estimates received from the variable nodes $\mathcal{I}_{nv,j}^c$ sum up to y'. Thus, in a worst case scenario check nodes must perform $d_v^{d_c}$ searches to ensure that no combination sums up to y'. Fortunately, we can exploit that most measurement are generated by a reduced number of non-null components of vector $\mathbf{x}_0$ since both the edges and variable nodes represent the non-null entries of the signal $\mathbf{x}_0$ with probability γ. We refer to these edges and variable nodes as *active*. The number of active edges per check node follows a binomial distribution in the N asymptotic regime.

A fraction of the check nodes is generated by the contribution of a single non-null entry of the signal vector (with probability $\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{S}| = 1\big)$). Therefore, some variable nodes receive the same estimates from different check nodes so variable nodes can decide that the repeated estimate is their real value (if the graph is free of length four cycles or whenever the weights of the edges are i.i.d. according to some continuous distribution the values y' in equation 8.4 will be all different).

Thus, verification-based algorithms require a sufficiently large fraction of measurements equal to a single variable node $\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{S}| = 1\big)$ in order to enable the verification process at the variable nodes or to allow them to forward to the check nodes a list of estimates that includes the real value of the variable node.

Let us have a look of the decoding process. For the sake of clarity let us focus on a right regular code with check node of degree d_c. The decoding process starts whenever $\mathrm{P}\big(|\mathcal{I}^c \cap \mathcal{S}| = 1\big)$ is high enough to enable variable nodes to exploit the V3 decoding rule, i.e. to initiate the verification process.

In this case, the neighborhoods of check nodes contain in average a low number of non-null variable nodes since the amount of non-null components contributing to measurements follows a binomial distribution. So there are usually several variable nodes that represent *zeros* contributing to the generation of a given measurement. Due to this check nodes can initially perform the searches assuming that almost all the variable nodes in their neighborhood are zero, hoping to find a combination of estimates that sums up to the y' in equation (8.4).

In general, for irregular check node degree distributions, there are several mass concentrations but it also holds that the one belonging to the check nodes with the lowest degree is concentrated closer to $i = 1$, and the remaining mass of probability are concentrated at larger values. Hence, it is also very unlike to find the proper combination of estimates at the check nodes with larger number of edges.

Algorithm 8.5 in appendix 8.A.3 summarizes the check node operations. Basically, the j-check node removes from its measurement y_j the contribution of the verified variable nodes in its neighborhood as in equation (8.4). After that, similarly to the VA algorithm, if $y' = 0$ sends a verified message with the value y' to all the variable nodes indexed in $\mathcal{I}_{nv}^c$. Otherwise, executes an exhaustive search, testing whether any combination sums up to y'. Note that $\mathbf{X}_j$ represents the list of estimates received from the non-verified variable nodes that are indexed in $\mathcal{I}_{nv}^c$[1]. After performing the search, if any check nodes find a combination of estimates that sum up to y', they send to all the variable nodes indexed in $\mathcal{I}_{nv}^c$ a verified message with the proper value. Otherwise, as in the general VA, they send a non-verified message with an estimate y' to all the variable nodes indexed in $\mathcal{I}_{nv}^c$.

Algorithm 8.6 in appendix 8.A.3 summarizes variable node operation. In short, whenever variable nodes receive a verified message or at least two non-verified messages with the same estimate, their state swaps to verified with the value indicated. Otherwise, they keep in a non-verified state and append to the non-verified flag a list with all the estimates received though the remaining edges.

8.3.3 Computational complexity

In this section we evaluate the computational complexity of the proposed algorithm for a fixed sparsity ratio γ and rate r under the assumptions that the length N is large enough so that the signal vector has approximately γN nonzero entries. Furthermore, recall from section 3.1 that theoretically the number of samples M to almost perfectly reconstruct a γ-sparse signal of length N vector is $\mathrm{M} = \gamma\mathrm{N} + \mathrm{o}(\mathrm{N})$, hence M is $\mathrm{O}(\gamma\mathrm{N})$. We consider that comparisons, products and sums have the same complexity.

The computational complexity of the activation of a variable node is dominated by the search of at least two estimates with equal value and it requires $\binom{d_v}{2}$ operations. So the activation of all the variable nodes is $\mathrm{O}(d_v^2\mathrm{N})$.

1 This list has in average $|\mathcal{I}_{nv}^c| \sum_{\forall d_v} \mathrm{L}(d_v) d_v$ values. The wise way to perform the search is first assume that all but one input messages are zero, i.e. performs $(d_v - 1)\binom{|\mathcal{I}_{nv}^c|}{1}$ comparisons, then assume that two components are non null, i.e. perform $(d_v - 1)^2\binom{|\mathcal{I}_{nv}^c|}{2}$ comparisons, and so on.

The list of the messages that the variable and check nodes send to the check and variable nodes has at most length $d_v - 1$ and 1, respectively. These messages have to be multiplied/divided by the weight of the edges, so the complexity of this operation is $\mathrm{O}(d_v^2\mathrm{N})$, as there are $d_v\mathrm{N}$ edges in the graph.

Each check node performs at most $d_v^{d_c}$ comparisons. Hence, the activation of all the check nodes in one iteration has a computational complexity $\mathrm{O}(\gamma\mathrm{N}d_v^{d_c})$.

Thus, the overall computational complexity of the first iteration is $\mathrm{O}(\mathrm{N}(\gamma d_v^{d_c} + 2d_v^2))$. Further iterations require fewer computations because when the verification process starts the amount of verified variable nodes increases and so the amount of verified messages that check nodes receive also increases. This fact reduces the amount of computations that check nodes perform to search if any combination of estimates is equal to its measure.

8.3.4 Performance

Figure 8.2 depicts the performance of the NB-VA and CS-LMP algorithm for different sequence lengths compared with the VA performance threshold (VA-threshold) and BP-threshold of the graph ensemble[1]. The horizontal axis represents the sampling efficiency (K/M) and the vertical axis the sequence error rate (SER), $\mathrm{P}(\mathbf{x}_0 \neq \mathrm{g}(\mathrm{q}(\mathbf{x}_0)))$. The plots of the performance of the NB-VA and CS-LMP were obtained by Monte Carlo simulations. Each simulation was stopped when 200, 200 or 20 erroneous reconstructions were obtained, for $\mathrm{N} = 2000$, 10000 and 100000, respectively. A graph degree profile optimized in [53] for the NB-VA of rate 0.8 was employed in all simulations, it included two degrees for check and variable nodes and the matrix coefficients were either 0 or 1. The non-null entries of the $\mathbf{x}_0$ are i.i.d. according to a standard distribution. Note that verification algorithms declare that a sequence is perfectly reconstructed whenever all variable nodes are verified and thus, the values of the entries of estimate vector are equal to the ones of the original sequence.

Figure 8.2 shows that both the performance of the NB-VA and CS-LMP increases as N increases, as expected. It also shows that the performance of the NB-VA approaches VA-threshold obtained by density evolution in [53] as N increases. The figure also shows that the CS-LMP clearly outperforms the NB-VA for all N, being the gain of efficiency approximately of 0.07 irrespective of N. In spite of this gain, the gap to BP-threshold of the graph ensemble is still large.

Figure 8.3 compares the performance of the CS-LMP with the one of the estimate-maximize Gaussian-Bernoulli Approximate Message Passing (EM-GB-AMP) algorithm [153] and the BP-threshold, for different block lengths. The entries of the measurement matrix of the EM-GB-AMP

1 See section 3.3.2 for further detail on BP-threshold and section 3.2.5 for more information on VA-thresholds.

are i.i.d according to a standard distribution. The measurement matrix of the CS-LMP is the same sparse matrix of the simulations of figure 8.2 (from [53]). Each plot of the figure is obtained for a fixed matrix and changing the sparsity γ. Note that both the Gaussian and the sparse matrix have the same rate.

The horizontal axis represents the sampling efficiency and the vertical axis the SER. The performance plots of the EM-GB-AMP and CS-LMP were obtained by Monte Carlo simulations and each simulation was stopped once 200 erroneous reconstructions were obtained. As before, for the LMP a sequence is declared perfectly reconstructed whenever all variable nodes are verified. For the EM-GB-AMP, a sequence is declared free of errors whenever the normalized minimum square error is $\frac{\|\mathbf{x}_0-\mathbf{x}\|_2^2}{\|\mathbf{x}_0\|_2^2} < 10^{-4}$.

Figure 8.3 shows that for the shortest block length, N = 2000, 10000, initially the EM-GB-AMP algorithm outperforms the CS-LMP algorithm but as the signal becomes more sparse, the SER of the EM-GB-AMP algorithm shows an error floor whereas the CS-LMP SER keeps decreasing and eventually the CS-LMP outperforms the EM-GB-AMP.

For the largest block length, both algorithms (with their own different measurement matrices) have the same performance. Notice that for this block length, the plot of the EM-GB-AMP algorithm also seems to depict an error floor.

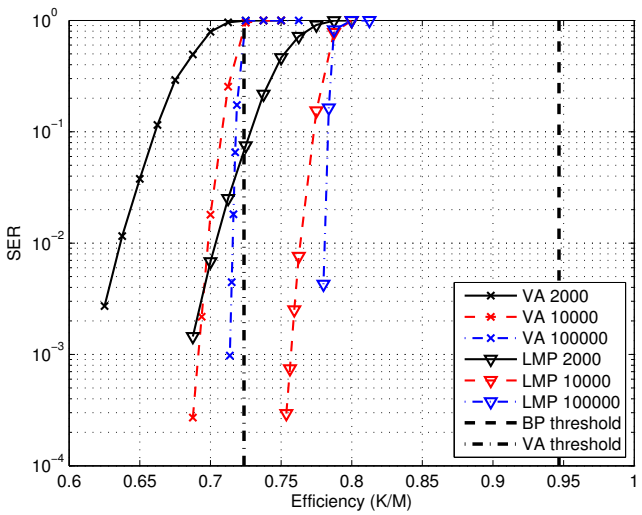


Figure 8.2: Performance comparison of CS-LMP (LMP) and NB-VA (VA) and different N compared to the thresholds of the NB-VA and the BP-threshold of the graph ensemble.

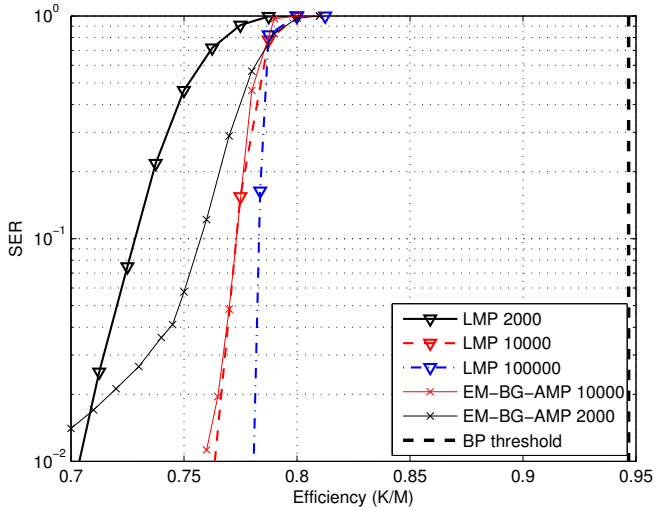


Figure 8.3: Performance comparison of CS-LMP (LMP) vs. EM-GB-AMP for different N compared to the threshold of the BT-threshold of the ensemble of graphs.

Appendix 8.A Algorithms

8.A.1 Verification Algorithm

Algorithm set up

The NB-verification algorithm, referred to as simply as verification algorithm, is an iterative message passing strategy on a sparse graph representation of a matrix with {0,1} coefficients. The message passing strategy is divided in iterations where check and variable nodes are activated alternatively.

Messages exchanged between check and variable nodes have to fields {'flag','value'}. The field 'value' is either an estimate of the value of the component that the variable node represents to. The flag field can be either 'v' to indicate that the value is verified, 'nv' to indicate that the value (if any) is non-verified.

Check node update rules for VA

Algorithm 8.1 describes the check node update rules for of VA. Its goal is to generate the messages that check nodes send to variable nodes.

The input arguments of the algorithms are all the messages generated by the variable nodes, enclosed in $\mathbf{X}_{v\to c}$, the connexions of the graph and their weights (coefficients of the matrix), enclosed in $\left\{\mathcal{J}_1^c,\ldots,\mathcal{J}_j^c,\ldots,\mathcal{J}_{\mathrm{M}}^c\right\}$, and the vector of measures $\mathbf{y}$. Note that each set $\mathcal{J}^c$ encloses the

connections and the weights of the edges departing from the check node. The output arguments of the algorithm are the messages generated by the checks and sent to the variable nodes $\mathbf{X}_{v\to c}$.

input : $\mathbf{y}$, $\mathbf{X}_{v\to c}$, $\left\{\mathcal{I}_1^c,\ldots,\mathcal{I}_j^c,\ldots,\mathcal{I}_{\mathrm{M}}^c\right\}$
output : $\mathbf{X}_{v\to c}$

```
1  for i ← 1 to M do
2      y ← y_i ;
3      cont ← |I_i^c| ;
4      I ← I_i^c ;
5      for j ∈ I_i^c do
6          if flag_j = v then
7              y' ← y' − value_j ;
8              cont ← cont − 1 ;
9              Remove(I, j);
10         end
11     end
12     if  y=0 then PropagateMsg(I,v, 0);
13     else if cont = 1 then PropagateMsg(I,v, y);
14     else PropagateMsg(I,nv, y);
15 end
```

Algorithm 8.1: Check node update rules for Verification Algorithm for Compressed sensing.

In the first 'for' loop (line 1) the algorithm visits the M check nodes and generates a temporal copy of the data. The second 'for' loop (line 5) is aimed to remove the contribution of the verified messages. Finally, in lines 12 to 14 the algorithm updates the output messages.

The reader should note that the function 'Remove' in line 9 removes from the data structure $\mathcal{I}$ the references to the variable node j. Finally, the function 'Propagate' updates all the output positions of the structure $\mathbf{X}_{v\to c}$ indicated by the first argument, i.e. $\mathcal{I}$. The second argument indicates the flag of all the output messages indicated by $\mathcal{I}$ and the third argument indicates the value of all output messages. Note that the output value of the message will be equal to $x'_{c_i\to v_j} = y/a_{i,j}$ where $a_{i,j}$ is the corresponding coefficient of the measurement matrix.

Variable node update rules for VA

Algorithm 8.2 describes a basic update rules for variable nodes of VA. Its goal is to generate the messages that variable nodes send to check nodes.

The input arguments of the algorithms are all the messages generated by the variable nodes, enclosed in $\mathbf{X}_{c\to v}$, the connexions of the graph and their weights (coefficients of the matrix), enclosed in $\left\{\mathcal{I}_1^v,\ldots,\mathcal{I}_j^v,\ldots,\mathcal{I}_{\mathrm{N}}^v\right\}$, and the states of each variable node $\{V_1,\ldots,V_j,\ldots,V_{\mathrm{N}}\}$. Note that each $\mathcal{I}$ encloses the connections and the weights of the edges departing from the variable node. The output arguments of the algorithm are the messages generated by the checks and sent to the variable nodes $\mathbf{X}_{v\to c}$.

input : $\mathbf{X}_{c\to v}$, $\left\{\mathcal{I}_1^v,\ldots,\mathcal{I}_j^v,\ldots,\mathcal{I}_{\mathrm{N}}^v\right\}$, $\{V_1,\ldots,V_j,\ldots,V_{\mathrm{N}}\}$
output: $\mathbf{X}_{v\to c}$

1 **for** $i \leftarrow 1$ **to** N **do**
2 **for** $\forall j \in \mathcal{I}_i^v$ **do**
3 x $\leftarrow \mu_{j\to i}(value)$;
4 **if** *flag* = *'v'* **then** PropagateMsg($\mathcal{I}_i^v$, v,x);
5 **else if** CompareMsg($\mathcal{I}_i^v \setminus j$,$\mathbf{X}_{c\to v}^i$,$x$)= *true* **then**
6 PropagateMsg($\mathcal{I}_i^v$, v,x)
7 **end**
8 **end**
9 **if** $V_i ='$ nv$'$ **then**
10 PropagateMsg($\mathcal{I}_i^v$, nv,$-$)
11 **end**
12 **end**

Algorithm 8.2: Variable node update rules for Verification Algorithm for noiseless Compressed sensing.

The first 'for' loop (line 1) processes sequentially all the N variable nodes. Then the second 'for' loop (line 2 updates each variable to check message. Line 4 handles the case that an incoming message is verified, i.e. it forwards this message to all the other check nodes in the neighborhood of the variable node and updates the state of the variable node (function 'Propagate'). Line 5 handles the other verification case. Function 'Compares' perform compares the value of the message received through the j-th edge with the messages received by the other edges and outputs a 'true' flag when finds a coincidence. Then the function 'Propagate' propagates this message to the remaining check nodes of the neighborhood and updates the state of the variable node.

Finally, line 9 handles the case when the state of the variable node is still 'nv'. In this case the variable node propagate to all the check nodes in its neighborhood a message 'nv' with a dummy value.

8.A.2 Enhanced verification algorithm

The enhanced verification algorithm is an iterative message passing strategy on a sparse graph representation of a matrix with {0,1} coefficients. The message passing strategy is divided in iterations where check and variable nodes are activated alternatively.

Messages exchanged between check and variable nodes have to fields {'flag','value'}. The field 'value' is either an estimate of the value of the component that the variable node represents to. The flag field can be either 'v' to indicate that the value is verified, 'nv' to indicate that the value (if any) is non-verified and 'c' to indicate that variable nodes detected that two or more check nodes sent messages with the same value.

Check node update rules for EVA

Algorithm 8.3 describes a basic update rules for check nodes of VA. Its goal is to generate the messages that check nodes send to variable nodes

input : $\mathbf{y}$, $\mathbf{X}_{v\to c}$, $\left\{\mathcal{J}_1^c,\ldots,\mathcal{J}_j^c,\ldots,\mathcal{J}_{\mathrm{M}}^c\right\}$
output : $\mathbf{X}_{v\to c}$

1 **for** $i \leftarrow 1$ **to** M **do**
2 $y \leftarrow y_i$;
3 $\mathcal{J}_{nv} \leftarrow \mathcal{J}_i^c$;
4 $\mathcal{J}_c \leftarrow \mathcal{J}_i^c$;
5 **for** $j \in \mathcal{J}_i^c$ **do**
6 **if** $\mathrm{flag}_j = v$ **then** ;
7 $y' \leftarrow y' - value_j$;
8 Remove($\mathcal{J}_{nv}$, j);
9 Remove($\mathcal{J}_c$, j);
10 **else if** $\mathrm{flag}_j = c$ **then** ;
11 Remove($\mathcal{J}_{nv}$, j);
12 **else** Remove($\mathcal{J}_c$, j);
13 **end**
14 **if** $y{=}0$ **then** PropagateMsg($\mathcal{J}_{nv}$,v, 0);
15 **else if** $|\mathcal{J}_{nv}|{=}1$ & $|\mathcal{J}_c|{=}0$ **then** PropagateMsg($\mathcal{J}_{nv}$,v, y);
16 **else if** $|\mathcal{J}_{nv}|{>}1$ & $|\mathcal{J}_c|{=}0$ **then** PropagateMsg($\mathcal{J}_{nv}$,nv, y);
17 **else if** $|\mathcal{J}_c|{>}0$ **then** ;
18 PropagateMsg($\mathcal{J}_{nv}$,v, 0);
19 PropagateMsg($\mathcal{J}_c$,nv, y) ;
20 **end**

Algorithm 8.3: Check node update rules for enhanced verification Algorithm for Compressed sensing.

The input arguments of the algorithm are all the messages generated by the variable nodes, enclosed in $\mathbf{X}_{v\to c}$, the connexions of the graph and their weights (coefficients of the matrix), enclosed in $\left\{\mathcal{J}_1^c,\ldots,\mathcal{J}_j^c,\ldots,\mathcal{J}_{\mathrm{M}}^c\right\}$, and the vector of measures $\mathbf{y}$. Note that each $\mathcal{J}$ encloses the connections and the weights of of the edges departing from the check node. The output arguments of the algorithm are the messages generated by the checks and sent to the variable nodes $\mathbf{X}_{v\to c}$.

In the first 'for' loop (line 1) the algorithm basically visits the M check nodes, generates a temporal copy of the data.

The second 'for' loop (line 5) is aimed to remove the contribution of the messages whose state is verified, partitioning the edges in three sub-sets depending of the received message: coincident edges $\mathcal{I}_c$, non-verified edges $\mathcal{I}_{nv}$ and the verified ones (not taken into account).

Finally in lines 14 to 17 the algorithm performs update the output messages.

The reader should note that the function 'Remove' in line 6 removes from the data structure $\mathcal{I}$ the references to the variable node j. Finally the function 'Propagate' updates all the output positions of the structure $\mathbf{X}_{v\to c}$ indicated by the first argument, i.e. $\mathcal{I}$. The second argument indicates the flag of all the output messages indicated by $\mathcal{I}$ and the third argument indicates the value of all output messages. Note that the output value of the message will be equal to $x'_{c_i\to v_j} = y$, since now the matrix is binary.

Variable node update rules for EVA

Algorithm 8.4 describes a basic update rules for variable nodes of VA. Its goal is to generate the messages that variable nodes send to check nodes.

input : $\mathbf{X}_{c\to v}$, $\left\{\mathcal{I}_1^v,\ldots,\mathcal{I}_j^v,\ldots,\mathcal{I}_{\mathrm{N}}^v\right\}$, $\{V_1,\ldots,V_j,\ldots,V_{\mathrm{N}}\}$
output: $\mathbf{X}_{v\to c}$

for $i \leftarrow 1$ **to** N **do**
  $\mathcal{I}_v = \emptyset$;
  $\mathrm{x}_v = 0$;
  $[\mathcal{I}_c, x_c]$=CompareMsg($\mathcal{I}_i^v$,$\mathbf{X}_{c\to v}^i$) ;
  for $\forall j \in \mathcal{I}_i^v$ **do**
    $\mathrm{x} \leftarrow \mu_{j\to i}(value)$;
    flag $\leftarrow \mu_{j\to i}(flag)$;
    if $flag = v$ **then**
      add($\mathcal{I}_i^v \setminus j$,$\mathcal{I}_v$);
      $\mathrm{x}_v \leftarrow \mathrm{x}$
    end
  end
  if $|\mathcal{I}_v| > 0$ **then**
    PropagateMsg($\mathcal{I}_v$,v,x_v);
    PropagateMsg($\mathcal{I}_i^v \setminus \mathcal{I}_v$,nv,$-$);
    set(V_i,v,x_v)
  end
  else if $|\mathcal{I}_c| > 0$ **then**
    PropagateMsg($\mathcal{I}_c$, c,x_c);
    PropagateMsg($\mathcal{I}_i^v \setminus \mathcal{I}_c$, nv,$-$);
    set(V_i,nv,$-$)
  end
  else
    PropagateMsg($\mathcal{I}_i^v$, nv,$-$);
    set(V_i,nv,$-$)
  end
end

Algorithm 8.4: Variable node update rules for Enhanced Verification Algorithm for noiseless Compressed sensing.

The input arguments of the algorithms are all the messages generated by the check nodes,

enclosed in $\mathbf{X}_{c\to v}$, the connexions of the graph and their weights (coefficients of the matrix), enclosed in $\left\{\mathcal{I}_1^v,\ldots,\mathcal{I}_j^v,\ldots,\mathcal{I}_{\mathrm{N}}^v\right\}$, and the states of each variable node $\{V_1,\ldots,V_j,\ldots,V_{\mathrm{N}}\}$. Note that each $\mathcal{I}$ encloses the connections and the weights of the edges departing from the variable node. The output arguments of the algorithm are the messages generated by the checks and sent to the variable nodes $\mathbf{X}_{v\to c}$.

The first 'for' loop (line 1) processes sequentially all the N variable nodes. The function 'CompareMsg' (in line 4) compares all the values of the messages received from the check nodes indexed in $\mathcal{I}_i^v$ and generates a set $\mathcal{I}_c$ with the coincident indexed messages and the coincident value x_c. After that in the second loop, line 5, the set of verified edges $\mathcal{I}_v$ is populated with according to the verification rules.

Finally, the function is ready to update the output messages $\mathbf{X}_{v\to c}$. At the first block (line 13) is triggered in case that the variable node received any verified message. In this case the messages to be propagated through the edges indexed in $\mathcal{I}_v$ are updated with the flag 'v' and the proper value x_v whereas the remaining edges are updated with the 'nv' flag. In this case the state of the variable node v_i is updated to verified with the value x_v. The next block (line 18) updates the coincident and non-verified messages and sets the state of the variable node to non-verified. Finally, in case that the variable node didn't detect any coincidence nor received any verified message it informs to all check nodes in its neighborhood that its state is still non-verified.

8.A.3 List message passing for compressed sensing

Algorithm set up

The list message passing algorithm for compressed sensing is an iterative message passing strategy on a sparse graph representation of a matrix with real valued coefficients. The message passing strategy is divided in iterations where check and variable nodes are activated alternatively.

Messages exchanged between check and variable nodes have to fields {'flag','value'}. When the 'flag' is 'v' the field 'value' is an estimate of the value of the component that the variable node represents to whereas when the flag is 'nv' the field 'value' is a list of estimates of the component.

Check node update rules for CS-LMP

The input arguments of the algorithm are the vector of measurements $\mathbf{y}$, a two-dimension array $\mathbf{X}_{v\to c}$ that contains the messages that variable nodes sent to each check nodes[1] and information

1 Each one of these messages recall that each one of this messages may be composed by an arbitrarily large amount of estimates.

about the connections of the graphs and the weights of the edges.

In this algorithm check nodes are processed sequentially. At the beginning of each iteration the algorithm stores a temporal copy of the indexes of the variable nodes in the neighborhood of the check node $\mathcal{I}_i^c$ and of its respective measurement. After that, in the second 'for' (see line 4) the contribution of the verified variable nodes is removed from the measurement y. In this loop the variable nodes connected to check node that sent a non-verified message are also identified. Note that function 'setCoeff' (see line 7) retrieves the coefficient of the measurement matrix.

After that, see line 13, the algorithm performs an exhaustive search. As stated in section 8.3.2, the search can be focused on the combinations that are more likely, i.e. we assume first that all but one of the components are zero, then all but two are zero and so on. Each time we evaluate all the possibilities. In case that the check node have a too large degree we can stop the exhaustive search after checking that no combination of 4 o 5 components of the list of estimates sums up to y' since it is quite unlikely the presence of correct value in several lists at the same time[1].

Once the search has ended, the check nodes forward to the variable nodes the results. In case that the search was successful they forward to each variable node the proper value of the combination $\hat{\mathbf{x}}$ along with a flag 'v', see line 14. Otherwise, they send to all the variable nodes indexed in $\mathcal{I}_{nv}$

1 Remind that the values x must be multiplied by the weight of the edge

a message with the value y along with a 'nv' flag[1].

input : $\mathbf{y}$, $\mathbf{X}_{v\to c}$, $\left\{\mathcal{I}_1^c,\ldots,\mathcal{I}_j^c,\ldots,\mathcal{I}_{\mathrm{M}}^c\right\}$
output : $\mathbf{X}_{v\to c}$

for $i \leftarrow 1$ **to** M **do**
  y$\leftarrow y_i$;
  $\mathcal{I}_{nv} \leftarrow \mathcal{I}_i^c$;
  for $\forall j \in \mathcal{I}_i^v$ **do**
    $x \leftarrow \mu_{j\to i}(value)$;
    flag $\leftarrow \mu_{j\to i}(flag)$;
    $a_{i,j} \leftarrow$ setCoeff(i,j) ;
    if *flag = 'v'* **then**
      $y \leftarrow y - a_{i,j}x$;
      $\mathcal{I}_{nv} \leftarrow \mathcal{I}_{nv} \setminus j$
    end
  end
  [successful, $\hat{\mathbf{x}}$] $\leftarrow$ ExhaustiveSearch(y,$\mathcal{I}_{nv}$,$\mathbf{X}_{v\to c}^i$) ;
  if *successful* **then**
    PropagateMsg($\mathcal{I}_{nv}$,$\hat{\mathbf{x}}$)
  end
  else
    PropagateMsg($\mathcal{I}_{nv}$,y)
  end
end

Algorithm 8.5: Check node update rules for List Message Passing for Compressed sensing.

1 Remind that $x_j = y_i/a_{i,j}$

Variable node update rules for CS-LMP

Algorithm 8.6 describes a basic update rules for variable nodes of VA. Its goal is to generate the messages that variable nodes send to check nodes.

input : $\mathbf{X}_{c\to v}$, $\left\{\mathcal{I}_1^v,\ldots,\mathcal{I}_j^v,\ldots,\mathcal{I}_{\mathrm{N}}^v\right\}$, $\{V_1,\ldots,V_j,\ldots,V_{\mathrm{N}}\}$
output: $\mathbf{X}_{v\to c}$

1 **for** $i \leftarrow 1$ **to** N **do**
2 **for** $\forall j \in \mathcal{I}_i^v$ **do**
3 x $\leftarrow \mu_{j\to i}(value)$;
4 **if** $flag = v$ **then** PropagateMsg($\mathcal{I}_i^v$, v,x);
5 **else if** CompareMsg($\mathcal{I}_i^v \setminus j$,$\mathbf{X}_{c\to v}^i$,$x$)$= true$ **then**
6 PropagateMsg($\mathcal{I}_i^v$, v,x)
7 **end**
8 **end**
9 **if** $V_i =$ nv **then**
10 PropExtrinMsg($\mathcal{I}_i^v$, nv,$\mathbf{X}_{c\to v}^i$)
11 **end**
12 **end**

Algorithm 8.6: Variable node update rules for Verification Algorithm for noiseless Compressed sensing.

The input arguments of the algorithms are all the messages generated by the check nodes, enclosed in $\mathbf{X}_{c\to v}$, the connexions of the graph and their weights (coefficients of the matrix), enclosed in $\left\{\mathcal{I}_1^v,\ldots,\mathcal{I}_j^v,\ldots,\mathcal{I}_{\mathrm{N}}^v\right\}$, and the states of each variable node $\{V_1,\ldots,V_j,\ldots,V_{\mathrm{N}}\}$. Note that each $\mathcal{I}$ encloses the connections and the weights of of the edges departing from the varable node. The output arguments of the algorithm are the messages generated by the checks and sent to the variable nodes $\mathbf{X}_{v\to c}$.

In this algorithm variable nodes are processed sequentially. At the beginning of each iteration the algorithm stores a temporal copy of the indexes of the variable nodes in the neighborhood of the check node $\mathcal{I}_i^c$ and of its respective measurement.

The first 'for' loop (line 1) processes sequentially all the N variable nodes. Then the second 'for' loop (line 2 updates each variable to check message. Line 4 handles the case that an incoming message is verified, i.e. it forwards this message to all the other check nodes in the neighborhood of the variable node and updates the state of the variable node (function 'Propagate'). Line 5 handles the other verification case. Function 'Compares' perform compares the value of the message received through the j-th edge with the messages received by the other edges and

outputs a 'true' flag when finds a coincidence. Then the function 'Propagate' propagates this message to the remaining check nodes of the neighborhood and updates the state of the variable node.

Finally, line 9 handles the case when the state of the variable node is still 'nv'. In this case the variable node propagates to each check node in its neighborhood a message 'nv' and appends the estimates received from all the other check nodes in its neighborhood, i.e. appends a list of messages of length $d_v - 1$. The function 'PropExtrinMsg' performs this operation.

PART IV

Conclusions

CHAPTER 9

Conclusions

This thesis has focused on investigating and developing novel compression schemes for binary source coding, noiseless compressed sensing and group testing within the framework of analog compression.

Our contributions for GT and CS are devoted to both the characterization of the encoder and development of message passing strategies for the decoder side. Our purpose throughout the thesis was to gain insight on the encoder functions for GT and CS and to exploit it at the decoder side.

Encoding architecture :

In this thesis the different proposed compression schemes are based on the same encoder scheme: a multistage structure which has proved to be well suited for dealing with sequential and/or adaptive compression schemes.

In section 4.4 we have proposed an adaptive hybrid LDPC-LDPrC based code for binary source compression. This was our first approach to construct adaptively matrices exploiting the partial information that the erasure decoder retrieves from the OR/AND-based check nodes. We have showed that matrix structure is beneficial for fixed-length binary source compression schemes because it enables both the introduction of some degree of adaptivity within the compression scheme and the design of a specially tailored graph for each source sequence $\mathbf{x}_0$, which results in a performance gain when dealing with low entropy sources.

In section 8.2 we have proposed to use this multistage structured matrix in a sequential compressed sensing scheme where samples are generated until the source sequence is perfectly reconstructed. With its introduction one can control when the components are sensed and, hence, it is possible to assign the sensing resources in a fairer way, e.g. we can

design the matrix to sense equally early all the components, which results in an increased sampling efficiency.

Finally, in chapter 5 we presented a novel adaptive group testing scheme that also employs the multistage matrix structure. In this case, tests are generated in a sequential block-wise process, exploiting the information gathered in the previous blocks to design the next piece of the matrix. This design helps to reduce the amount of edges connected to check nodes of type III increasing the sampling/testing efficiency of the overall adaptive scheme. This approach seems very promising within the group testing framework.

In this thesis two criteria have been proposed to design the check node degree profile of the different matrix stages. Both lead to closed-form expressions. One has been introduced in the context of the GT and the other one in CS, but both can be applied to both scenarios.

Group testing problem :

We have approached the GT problem from both a theoretical and a practical point view.

In chapter 4 our research efforts were focused on modeling the mapping capabilities of the group testing encoder in the N-asymptotic regime when the matrix belongs to an ensemble of sparse graphs. Within the N-asymptotic regime we showed that the mapping of the encoder is non-injective and we deduced an upper bound for the average amount of sequences generated by a binary source of data that are mapped to a single sequence. We also have computed the fraction of unlabeled subjects at the output the erasure decoder as a function of the matrix ensemble and the source statistics. This result motivated the proposal of the adaptive GT scheme to cope with the limitations of the non-adaptive GT scheme.

After analyzing the non-adaptive encoder, in chapter 5 we decided to follow a different approach to obtain a lossless reconstruction scheme for GT: perform the testing process in multiple stages in a block-wise sequential manner so as at the end of each stage the information available at the output of the decoder is employed to design the next testing stage in which only the subjects that remain to be labeled are involved.

In chapter 6 the design of this multistage scheme has been systematized as a decision process. Within this problem we have developed a tool that accurately predicts how the matrix design affects the decision process evolution in the N-asymptotic regime.

The purpose of this tool is two-folded: it is aimed at both assisting to the designer of the policy to asses its performance and, furthermore, it can assist to the agent executing the

actions to choose between a set of actions, since it can characterize the expected outcome of each action given the information available at the decoder side.

Noiseless compressed sensing :

We have approached the CS problem from the theoretical and the practical side of the problem.

Within the theoretical approach to the noiseless CS, in chapter 7 we derived a necessary and sufficient condition that guarantees that in a noiseless compressed sensing scheme composed by a measurement matrix and a l_0-based decoder achieves lossless reconstruction. As shown in 7, we have concluded that in the N-finite regime lossless reconstruction is possible whenever the number of rows of the measurement matrix is larger than the sparsity of the original signal vector K whenever the spark of the matrix is large enough ($\text{spark}(\mathbf{A}) > \text{K} + 1$).

Within the practical approach to the CS problem, in chapter 8 we have introduced two verification-based message passing algorithms over the graph representation of the matrix: the enhanced verification algorithm (EVA) a very low complexity algorithm aimed at dealing with very sparse signals that outperforms the other verification-based algorithms when dealing with very sparse signals in the N-finite regime.

The list message passing for compressed sensing algorithm is the best performing verification-based algorithm at the cost of an increased computational complexity. We showed that the noiseless compressed sensing scheme composed by a CS-LMP based decoder and a specially designed random graph has a similar performance than a GAMP-based decoder with a linear encoder implemented with a Gaussian matrix.

CHAPTER 10

Bibliography

[1] M. Akcakaya, J. Park, and Vahid Tarokh: 'A coding theory approach to noisy compressive sensing using low density frames'. In*IEEE Trans. on Signal Processing* (Nov. 2011), vol. 59(11): pp. 5369–5379 (cit. on pp. 31, 32).

[2] C. Aksoylar, G. Atia, and V. Saligrama: 'Sparse signal processing with linear and non-linear observations: A unified shannon theoretic approach'. In*IEEE Information Theory Workshop. (ITW 2013)*. Sept. 2013: pp. 1–5 (cit. on p. 15).

[3] C. Aksoylar and V. Saligrama: 'Information-theoretic bounds for adaptive sparse recovery'. In*IEEE International Symposium on Information Theory. (ISIT 2014)*. June 2014: pp. 1311–1315 (cit. on pp. 15, 16).

[4] M. Aldridge, L. Baldassini, and K. Gunderson: 'Almost separable matrices'. In*arXiv preprint arXiv:1410.1826* (2014), vol. (cit. on p. 15).

[5] M. Aldridge, L. Baldassini, and O. Johnson: 'Group testing algorithms: Bounds and simulations'. In*IEEE Trans. on Inform. Theory* (June 2014), vol. 60(6): pp. 3671–3687 (cit. on pp. 12, 15, 21, 68).

[6] T. Ancheta: 'Syndrome-source-coding and its universal generalization'. In*IEEE Trans. on Inform. Theory* (July 1976), vol. 22(4): pp. 432–436 (cit. on pp. 12, 13).

[7] L. Applebaum, S.D. Howard, S. Searle, and R. Calderbank: 'Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery'. In*Applied and Computational Harmonic Analysis* (2009), vol. 26(2): pp. 283–290 (cit. on pp. 36, 38).

[8] G.K. Atia and V. Saligrama: 'Boolean compressed sensing and noisy group testing'. In*IEEE Trans. on Inform. Theory* (Mar. 2012), vol. 58(3): pp. 1880–1901 (cit. on p. 15).

[9] W.U. Bajwa, R. Calderbank, and S. Jafarpour: 'Model selection: Two fundamental measures of coherence and their algorithmic significance'. In*Proceedings of the IEEE International Symposium on Information Theory. (ISIT 2010)*. June 2010: pp. 1568–1572 (cit. on p. 38).

[10] L. Baldassini, O. Johnson, and M. Aldridge: 'The capacity of adaptive group testing'. In*Proceedings of the IEEE International Symposium on Information Theory Proceedings. (ISIT2013)*. July 2013: pp. 2676–2680 (cit. on pp. 11, 15–17).

[11] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin: 'A simple proof of the restricted isometry property for random matrices'. In*Constructive Approximation* (3 2008), vol. 28. 10.1007/s00365-007-9003-x: pp. 253–263 (cit. on pp. 27, 36).

[12] E. Barillot, B. Lacroix, and D. Cohen: 'Theoretical analysis of library screening using a N-dimensional pooling strategy'. In*Nucleic acids research* (1991), vol. 19(22): pp. 6241–6247 (cit. on p. 15).

[13] D. Baron, S. Sarvotham, and R.G. Baraniuk: 'Bayesian compressive sensing via belief propagation'. In*IEEE Trans. on Signal Processing* (Jan. 2010), vol. 58(1): pp. 269–280 (cit. on pp. 9, 31, 32).

[14] R. Berinde, A.C. Gilbert, P. Indyk, H. Karloff, and M.J. Strauss: 'Combining geometry and combinatorics: A unified approach to sparse signal recovery'. In*Proceedings of the 46th Annual Allerton Conference on Communication Control and Computing, (Allerton 2008)*. Sept. 2008: pp. 798–805 (cit. on p. 37).

[15] T. Blumensath and M. Davies: 'Iterative thresholding for sparse approximations'. In*Journal of Fourier Analysis and Applications* (5 2008), vol. 14. 10.1007/s00041-008-9035-z: pp. 629–654 (cit. on p. 31).

[16] P.T. Boufounos and R.G. Baraniuk: '1-Bit compressive sensing'. In*Proceedings of the 42nd Annual Conference on Information Sciences and Systems. (CISS 2008)*. Mar. 2008: pp. 16–21 (cit. on pp. 7, 15).

[17] W.J. Bruno, D.J. Balding, E.H. Knill, D. Bruce, C. Whittaker, N. Dogget, R. Stalling, and D.C. Torney: 'Design of efficient pooling experiments'. In*Genomics* (1995), vol. 26: pp. 21–30 (cit. on p. 15).

[18] G. Caire, S. Shamai, A. Shokrollahi, and S. Verdu: 'Universal variable-length data compression of binary sources using fountain codes'. In*Proceedings of the IEEE Information Theory Workshop, (ITW 2004)*. Oct. 2004: pp. 123–128 (cit. on p. 13).

[19] G. Caire, S. Shamai, and S. Verdu: 'An efficient scheme for reliable error correction with limited feedback'. In*Proceedings of the IEEE International Symposium on Information Theory. (ISIT 2005)*. Sept. 2005: pp. 1521–1525 (cit. on p. 13).

[20] G. Caire, S. Shamai, and S. Verdu: 'Lossless data compression with error correcting codes'. In*Proceedings of the IEEE International Symposium on Information Theory. (ISIT 2003)*. June 2003: p. 22 (cit. on pp. 12, 13).

[21] G. Caire, S. Shamai, and S. Verdu: 'Noiseless data compression with low-density parity-check codes'. In*DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (2004), vol. 66: pp. 263–284 (cit. on p. 13).

[22] R. Calderbank, S. Howard, and S. Jafarpour: 'Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property'. In*IEEE Journal of Sel. Topics in Signal Process.* (Apr. 2010), vol. 4(2): pp. 358–374 (cit. on p. 36).

[23] R. Calderbank, S. Howard, and S. Jafarpour: 'Sparse reconstruction via the Reed-Muller Sieve'. In*Proceedings of the IEEE International Symposium on Information Theory. (ISIT 2010)*. June 2010: pp. 1973–1977 (cit. on p. 38).

[24] E. Candes, J. Romberg, and T. Tao: 'Stable signal recovery from incomplete and inaccurate measurements'. In*Communications on Pure and Applied Mathematics* (Mar. 2006), vol. 59(8): pp. 1207–1223 (cit. on p. 36).

[25] E.J. Candes: 'Compressive sampling'. In*Proceedings of the International Congress of Mathematicians*. Vol. 3. 2006: pp. 1433–145 (cit. on p. 35).

[26] E.J. Candes, J. Romberg, and T. Tao: 'Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information'. In*IEEE Trans. on Inform. Theory* (Feb. 2006), vol. 52(2): pp. 489–509 (cit. on p. 35).

[27] E.J. Candes and T. Tao: 'Decoding by linear programming'. In*IEEE Trans. on Inform. Theory* (Dec. 2005), vol. 51(12): pp. 4203–4215 (cit. on p. 27).

[28] E.J. Candes and T. Tao: 'Near-optimal signal recovery from random projections: universal encoding strategies?' In*IEEE Trans. on Inform. Theory* (Dec. 2006), vol. 52(12): pp. 5406–5425 (cit. on pp. 3–5, 27, 30, 35).

[29] E.J. Candes and T. Tao: 'Rejoinder: Dantzig selector, statistical estimation when p is much larger than n'. In*Annals of Statistics* (2007), vol. 35: pp. 2392–2404 (cit. on p. 30).

[30] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson: 'Randomness conductors and constant-degree lossless expanders'. In*Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. STOC '02. Montreal, Quebec, Canada: ACM, 2002: pp. 659–668 (cit. on p. 37).

[31] C.L. Chan, P.H. Che, S. Jaggi, and V. Saligrama: 'Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms'. In*Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing. (Allerton 2011)*. Sept. 2011: pp. 1832–1839 (cit. on pp. 15, 21).

[32] V. Chandar: *A negative result concerning explicit matrices with the restricted isometry property.* Tech. rep. 2008 (cit. on p. 37).

[33] C.M. Chen, Y.P. Chen, T.C. Shen, and J.K. Zao: 'On the optimization of degree distributions in LT code with covariance matrix adaptation evolution strategy'. In*Proceedings of the IEEE Congress on Evolutionary Computation. (CEC2010)*. IEEE. 2010: pp. 1–8 (cit. on p. 13).

[34] S.S. Chen, D.L. Donoho, and M.A. Saunders: 'Atomic decomposition by basis pursuit'. In*SIAM Journal on Computing* (1 Dec. 1998), vol. 20: pp. 33–61 (cit. on p. 31).

[35] Y. Cheng, J. Guo, and F. Zheng: 'A new randomized algorithm for group testing with unknown number of defective items'. In*Journal of Combinatorial Optimization* (2013), vol.: pp. 1–10 (cit. on pp. 11, 15, 16, 68, 69).

[36] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama: 'Graph-constrained group testing'. In*IEEE Trans. on Inform. Theory* (Jan. 2012), vol. 58(1): pp. 248–262 (cit. on p. 15).

[37] G. Cormode and S. Muthukrishnan: 'What's hot and what's not: tracking most frequent items dynamically'. In*ACM Trans. on Database Systems (TODS)* (2005), vol. 30(1): pp. 249–278 (cit. on p. 15).

[38] Thomas M Cover and Joy A Thomas: *Elements of information theory.* John Wiley & Sons, 2012 (cit. on p. 11).

[39] W. Dai and O. Milenkovic: 'Subspace pursuit for compressive sensing signal reconstruction'. In*IEEE Trans. on Inform. Theory* (May 2009), vol. 55(5): pp. 2230–2249 (cit. on p. 31).

[40] A. De Bonis, L. Gasieniec, and U. Vaccaro: 'Optimal two-stage algorithms for group testing problems'. In*SIAM Journal on Computing* (2005), vol. 34(5): pp. 1253–1270 (cit. on p. 16).

[41] K. Do Ba, P. Indyk, E. Price, and D.P. Woodruff: 'Lower bounds for sparse recovery'. In*Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms. (SODA 2010)*. Austin, Texas: Society for Industrial and Applied Mathematics, 2010: pp. 1190–1197 (cit. on p. 36).

[42] T.T. Do, Lu Gan, N. Nguyen, and T.D. Tran: 'Sparsity adaptive matching pursuit algorithm for practical compressed sensing'. In*Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers, (Asilomar 2008)*. Oct. 2008: pp. 581–587 (cit. on p. 31).

[43] David L. Donoho and Michael Elad: 'Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization'. In*Proceedings of the National Academy of Sciences* (2003), vol. 100(5): pp. 2197–2202 (cit. on pp. 3–5, 27, 29).

[44] D.L. Donoho: 'Compressed sensing'. In*IEEE Trans. on Inform. Theory* (Apr. 2006), vol. 52(4): pp. 1289–1306 (cit. on pp. 5, 27, 30, 35, 36).

[45] D.L. Donoho, M. Elad, and V.N. Temlyakov: 'Stable recovery of sparse overcomplete representations in the presence of noise'. In*IEEE Trans. on Inform. Theory* (Jan. 2006), vol. 52(1): pp. 6–18 (cit. on p. 35).

[46] D.L. Donoho, A. Javanmard, and A. Montanari: 'Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing'. In*IEEE Trans. on Inform. Theory* (Nov. 2013), vol. 59(11): pp. 7434–7464 (cit. on pp. 33, 38).

[47] D.L. Donoho, A. Maleki, and A. Montanari: 'Message passing algorithms for compressed sensing: I. motivation and construction'. In*Proceedings of the IEEE Information Theory Workshop. (ITW 2010)*. Jan. 2010: pp. 1–5 (cit. on p. 33).

[48] D.L. Donoho, A. Maleki, and A. Montanari: 'Message passing algorithms for compressed sensing: II. analysis and validation'. In*Proceedings of the IEEE Information Theory Workshop. (ITW 2010)*. Jan. 2010: pp. 1–5 (cit. on p. 33).

[49] D.L. Donoho and J. Tanner: 'Counting faces of randomly-projected polytopes when the projection radically lowers dimension'. In*J. of the American Mathematical Soc.* (2009), vol.: pp. 1–53 (cit. on pp. 30, 36, 123).

[50] D.L. Donoho and J. Tanner: 'Precise undersampling theorems'. In*Proceedings of the IEEE* (June 2010), vol. 98(6): pp. 913–924 (cit. on p. 123).

[51] R. Dorfman: 'The detection of defective members of large populations'. In*Ann. Math. Statist.* (Dec. 1943), vol. 14(4): pp. 436–440 (cit. on pp. 3, 14).

[52] J.C. Dumm: 'Global and asymptotic converge rate estimates for a class of projected gradient processes'. In*SIAM Journal on Control and Optimization* (19 1981), vol. 3: pp. 368–400 (cit. on p. 31).

[53] Y. Eftekhari, A.H. Banihashemi, and I. Lambadaris: 'Analysis and design of irregular graphs for node-based verification-based recovery algorithms in compressed sensing'. In*Proceedings of the IEEE International Symposium on Information Theory (ISIT 2012)*. July 2012: pp. 1872–1876 (cit. on pp. 35–37, 128, 129).

[54] Y. Eftekhari, A. Heidarzadeh, A.H. Banihashemi, and I. Lambadaris: 'Density evolution analysis of node-based verification-based algorithms in compressed sensing'. In*IEEE Trans. on Inform. Theory* (Oct. 2012), vol. 58(10): pp. 6616–6645 (cit. on pp. 35, 36).

[55] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky: 'A wide-angle view at iterated shrinkage algorithms'. In*in SPIE (Wavelet XII*. 2007: pp. 26–29 (cit. on p. 31).

[56] P. Elias: 'Predictive coding–I'. In*Information Theory, IRE Transactions on* (Mar. 1955), vol. 1(1): pp. 16–24 (cit. on p. 12).

[57] P. Elias: 'Universal codeword sets and representations of the integers'. In*IEEE Trans. on Inform. Theory* (Mar. 1975), vol. 21(2): pp. 194–203 (cit. on p. 12).

[58] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright: 'Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems'. In*IEEE Journal of Selected Topics in Signal Processing* (Dec. 2007), vol. 1(4): pp. 586–597 (cit. on p. 31).

[59] A.K. Fletcher, S. Rangan, and V.K. Goyal: 'Necessary and sufficient conditions for sparsity pattern recovery'. In*IEEE Trans. on Inform. Theory* (Dec. 2009), vol. 55(12): pp. 5758–5772 (cit. on pp. 3, 7, 14).

[60] M Fornasier and R. Rauhut: 'Iterative thresholding algorithms'. In*Applied and Computational Harmonic Analysis* (2008), vol. 25(2): pp. 187–208 (cit. on p. 31).

[61] B.R. Gaines: 'Stochastic computing systems'. In*Advances in information systems science*. Springer, 1969: pp. 37–172 (cit. on pp. 16, 44).

[62] J. Garcia-Frias: 'Compression of correlated binary sources using turbo codes'. In*IEEE Communications Letters* (Oct. 2001), vol. 5(10): pp. 417–419 (cit. on p. 53).

[63] J. Garcia-Frias and Y Zhao: 'Compression of binary memoryless sources using punctured turbo codes'. In*IEEE Communications Letters* (Sept. 2002), vol. 6(9): pp. 394–396 (cit. on p. 53).

[64] A.C. Gilbert, M.A. Iwen, and M.J. Strauss: 'Group testing and sparse signal recovery'. In*Proceedings of the 42nd Asilomar Conference on Signals Systems and Computers. (ASILOMAR 2008)*. Oct. 2008: pp. 1059–1063 (cit. on p. 15).

[65] A.C. Gilbert, Y. Li, E. Porat, and M.J. Strauss: 'Approximate sparse recovery: optimizing time and measurements'. In*Proceedings of the 42nd ACM symposium on Theory of computing*. Cambridge, Massachusetts, USA: ACM, 2010: pp. 475–484 (cit. on p. 31).

[66] A.A. Goldstein: 'Convex programming in Hilbert spaces'. In*Bulletin of the American Mathematical Society* (1964), vol. 70: pp. 709–710 (cit. on p. 31).

[67] M.T. Goodrich and D.S. Hirschberg: 'Improved adaptive group testing algorithms with applications to multiple access channels and dead sensor diagnosis'. In*Journal of Combinatorial Optimization* (2008), vol. 15(1): pp. 95–121 (cit. on p. 15).

[68] A. Gupta and S. Verdu: 'Nonlinear sparse-graph codes for lossy compression'. In*IEEE Trans. on Inform. Theory* (May 2009), vol. 55(5): pp. 1961–1975 (cit. on pp. 13, 18).

[69] A. Gupta and S. Verdú: 'Nonlinear sparse-graph codes for lossy compression of discrete non-redundant sources'. In*Proceedings of the IEEE Information Theory Workshop. (ITW 2007)*. Sept. 2007: pp. 541–546 (cit. on pp. 13, 18).

[70] Yao-Win H. and A. Scaglione: 'On multiple access for distributed dependent sources: a content-based group testing approach'. In*Proceedings of the IEEE Information Theory Workshop, 2004. (ITW 2004)*. Oct. 2004: pp. 298–303 (cit. on p. 15).

[71] J. Hagenauer, J. Barros, and A. Schaefer: 'Lossless turbo source coding with decremental redundancy'. In*Proceedings of the International ITG Conference on Source and Channel Coding*. 2004: pp. 333–339 (cit. on p. 13).

[72] H. Hassanieh, P. Indyk, D. Katabi, and E. Price: 'Simple and practical algorithm for sparse Fourier transform'. In*Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2012: pp. 1183–1194 (cit. on p. 27).

[73] J.D. Haupt, R.G. Baraniuk, R.M. Castro, and R.D. Nowak: 'Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements'. In*Proceedings of the Forty-Third Asilomar Conference on Signals, Systems and Computers. (ASILOMAR 2009)*. Nov. 2009: pp. 1551–1555 (cit. on p. 16).

[74] E.S. Hong and R.E. Ladner: 'Group testing for image compression'. In*IEEE Trans. on Image Processing* (Aug. 2002), vol. 11(8): pp. 901–911 (cit. on p. 15).

[75] D.A. Huffman: 'A method for the construction of minimum-redundancy codes'. In*Proceedings of the IRE* (Sept. 1952), vol. 40(9): pp. 1098–1101 (cit. on pp. 3, 12).

[76] J.K. Hunter and B. Nachtergaele: *Applied analysis*. World Scientific Pub Co Inc, 2001 (cit. on pp. 103, 105, 106).

[77] F.K. Hwang: 'A method for detecting all defective members in a population by group testing'. In*Journal of the American Statistical Association* (1972), vol. 67(339): pp. 605–608 (cit. on pp. 11, 15, 16, 68, 69).

[78] P. Indyk: 'Deterministic superimposed coding with applications to pattern matching'. In*Proceedings of the 38th Annual Symposium on Foundations of Computer Science. (1997)*. Oct. 1997: pp. 127–136 (cit. on p. 15).

[79] P. Indyk, E. Price, and D.P. Woodruff: 'On the power of adaptivity in sparse recovery'. In*Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*. IEEE. 2011: pp. 285–294 (cit. on p. 16).

[80] S. Jafarpour: 'Deterministic compressed sensing'. In*Princeton University Thesis* (2011), vol. Ed. by Princeton University Thesis (cit. on p. 38).

[81] S. Jafarpour, X. Weiyu, B. Hassibi, and R. Calderbank: 'Efficient and robust compressed sensing using optimized expander graphs'. In*IEEE Trans. on Inform. Theory* (Sept. 2009), vol. 55(9): pp. 4299–4308 (cit. on p. 37).

[82] S. Jafarpour, R. Willett, M. Raginsky, and R. Calderbank: 'Performance bounds for expander-based compressed sensing in the presence of Poisson noise'. In*Proceedings of the Forty-Third Asilomar Conference on Signals, Systems and Computers. (Asilomar 2009)*. Nov. 2009: pp. 513–517 (cit. on p. 36).

[83] A. Javanmard and A. Montanari: 'State evolution for general approximate message passing algorithms, with applications to spatial coupling'. In*Information and Inference* (2013), vol. 2(2): pp. 115–144 (cit. on pp. 33, 38).

[84] S. Ji, Y. Xue, and L. Carin: 'Bayesian compressive sensing'. In*IEEE Trans. on Signal Processing* (June 2008), vol. 56(6): pp. 2346–2356 (cit. on pp. 31, 32).

[85] O. Johnson: 'Strong converses for group testing in the finite blocklength regime'. In*arXiv preprint arXiv:1509.06188* (2015), vol. (cit. on p. 15).

[86] L. P. Kaelbling, M.L. Littman, and A.R. Cassandra: 'Planning and acting in partially observable stochastic domains'. In*Artificial intelligence* (1998), vol. 101(1): pp. 99–134 (cit. on p. 74).

[87] A.B. Kahng and S. Reda: 'New and improved BIST diagnosis methods from combinatorial group testing theory'. In*IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* (2006), vol. 25(3): pp. 533–543 (cit. on pp. 15, 16).

[88] U. Kamilov, S. Rangan, M. Unser, and A. Fletcher: 'Approximate message passing with consistent parameter estimation and applications to sparse learning'. In*Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger. Curran Associates, Inc., 2012: pp. 2438–2446 (cit. on pp. 27, 33).

[89] W. Kautz and R. Singleton: 'Nonrandom binary superimposed codes'. In*IEEE Trans. on Inform. Theory* (Oct. 1964), vol. 10(4): pp. 363–377 (cit. on p. 15).

[90] T. Kealy, O. Johnson, and R. Piechocki: 'The capacity of non-identical adaptive group testing'. In*Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton 2014)*. Sept. 2014: pp. 101–108 (cit. on p. 15).

[91] M.A. Khajehnejad, A.G. Dimakis, Weiyu Xu, and B. Hassibi: 'Sparse recovery of nonnegative signals with minimal expansion'. In*IEEE Trans. on Signal Processing* (Jan. 2011), vol. 59(1): pp. 196–208 (cit. on p. 30).

[92] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová: 'Statistical-physics-based reconstruction in compressed sensing'. In*Physical Review X* (Apr. 2012), vol. 2(2): p. 021005 (cit. on pp. 34, 38).

[93] S. Kudekar and H.D. Pfister: 'The effect of spatial coupling on compressive sensing'. In*Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing. (Allerton 2010)*. 2010: pp. 347–353 (cit. on pp. 36, 38).

[94] C.H. Li: 'A sequential method for screening experimental variables'. In*Journal of the American Statistical Association* (1962), vol. 57(298): pp. 455–477 (cit. on p. 15).

[95] A.D. Liveris, Zixiang Xiong, and C.N. Georghiades: 'Compression of binary sources with side information at the decoder using LDPC codes'. In*IEEE Communications Letters* (Oct. 2002), vol. 6(10): pp. 440–442 (cit. on pp. 13, 53, 57).

[96] M. Luby: 'LT codes'. In*Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science. (2002)*. 2002: pp. 271–280 (cit. on pp. 13, 59).

[97] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman: 'Efficient erasure correcting codes'. In*IEEE Trans. on Inform. Theory* (Feb. 2001), vol. 47(2): pp. 569–584 (cit. on p. 36).

[98] G. Ma J.and Plonka and M.Y. Hussaini: 'Compressive video sampling with approximate message passing decoding'. In*IEEE Trans. on Circuits and Systems for Video Technology* (Sept. 2012), vol. 22(9): pp. 1354–1364 (cit. on p. 27).

[99] D.J.C. MacKay: 'Fountain codes'. In*IEEE Proceedings on Communications* (Dec. 2005), vol. 152(6): pp. 1062–1068 (cit. on pp. 13, 59).

[100] D.J.C. MacKay and R.M. Neal: 'Near Shannon limit performance of low density parity check codes'. In*Electronics Letters* (Aug. 1996), vol. 32(18): p. 1645 (cit. on p. 12).

[101] A. Maleki and D.L. Donoho: 'Optimally tuned iterative reconstruction algorithms for compressed sensing'. In*IEEE Journal of Selected Topics in Signal Processing* (Apr. 2010), vol. 4(2): pp. 330–341 (cit. on p. 31).

[102] S. Mallat and Z. Zhang: 'Matching pursuit with time-frequency dictionaries'. In*IEEE Trans. on Signal Processing* (1993), vol. 41: pp. 3397–3415 (cit. on p. 31).

[103] M. Malyutov: 'Information Theory, combinatorics and search theory,' in*Lecture Notes in Computer Science*. Springer, 2013 (cit. on p. 15).

[104] M. Malyutov: 'Mathematical models and results in the theory of screening experiments'. In*Theoretical Problems of Experimental Design, Soviet Radio, Moscow* (1977), vol.: pp. 5–69 (cit. on pp. 15, 19).

[105] D. Matas, M. Lamarca, and J. Garcia-Frias: 'Non-linear graph-based codes for joint source-channel coding'. In*Proceedings of the 6th International Symposium on Turbo Codes and Iterative Information Processing. (ISTC 2010)*. Sept. 2010: pp. 384–388 (cit. on pp. 14, 15, 17, 25, 56).

[106] D. Matas, M. Lamarca, and J. Garcia-Frias: 'Non-linear graph-based codes for source coding'. In*Proceerdings of the IEEE Information Theory Workshop. (ITW 2009) IEEE*. Oct. 2009: pp. 318–322 (cit. on pp. 6, 14, 15, 17, 19, 25, 55).

[107] C. Measson, A. Montanari, and R. Urbanke: 'Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding'. In*IEEE Trans. on Inform. Theory* (2008), vol. 54(12): pp. 5277–5307 (cit. on p. 17).

[108] N. Meinshausen and B. Yu: 'Lasso-type recovery of sparse representations for high-dimensional data'. In*The Annals of Statistics* (2009), vol.: pp. 246–270 (cit. on pp. 3, 7, 14).

[109] A. Montanari: 'Graphical models concepts in compressed sensing'. In*Compressed Sensing: Theory and Applications* (2012), vol.: pp. 394–438 (cit. on p. 31).

[110] B. K. Natarajan: 'Sparse approximate solutions to linear systems'. In*SIAM journal on computing* (2 Apr. 1995), vol. 24: pp. 227–234 (cit. on pp. 3, 7, 14).

[111] D. Needell and J.A. Tropp: 'CoSaMP: Iterative signal recovery from incomplete and inaccurate samples'. In*Applied and Computational Harmonic Analysis* (2009), vol. 26(3): pp. 301–321 (cit. on p. 31).

[112] D. Needell and R. Vershynin: 'Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit'. In*Foundations of Computational Mathematics* (3 2009), vol. 9. 10.1007/s10208-008-9031-3: pp. 317–334 (cit. on p. 31).

[113] M.L. Puterman: *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014 (cit. on p. 74).

[114] M. Raginsky, R.M. Willett, Z.T. Harmany, and R.F. Marcia: 'Compressed sensing performance bounds under poisson noise'. In*IEEE Trans. on Signal Processing,* (Aug. 2010), vol. 58(8): pp. 3990–4002 (cit. on p. 36).

[115] F. Ramirez-Javega and M. Lamarca: 'Variable-length source compression using successive refinement and non-linear graph-based codes'. In*Proceedings of the Data Compression Conference (DCC 2011)* (2011), vol.: p. 473 (cit. on p. 16).

[116] F. Ramirez-Javega, M. Lamarca, and J. Garcia-Frias: 'Progressive encoding with non-linear source codes for compression of low-entropy sources'. In*Proceedings of the 6th International Symposium on Turbo Codes and Iterative Information Processing. (2010)*. Sept. 2010: pp. 389–393 (cit. on pp. 19, 21, 41, 52).

[117] F. Ramirez-Javega, M. Lamarca, and J. Villares: 'Binary graphs and message passing strategies for Compressed Sensing in the noiseless setting'. In*Proceedings of the IEEE International Symposium on Information Theory Proceedings. (ISIT 2012)*. July 2012: pp. 1867–1871 (cit. on pp. 35, 38, 118, 121).

[118] F. Ramirez-Javega and M. Lamarca-Orozco: 'List message passing for compressed sensing'. In*Proceedings of the Signal Information Processing Association Annual Summit and Conference (APSIPA ASC),* Dec. 2015: pp. 1–4 (cit. on p. 125).

[119] F. Ramirez-Javega, D. Matas-Navarro, and M. Lamarca: 'Adaptive sampling for fast sparsity pattern recovery'. In*Proceedings of the 19th European Signal Processing Conference. (EUSIPCO 2011)*. Sept. 2011: pp. 348–352 (cit. on pp. 15, 16, 59, 63).

[120] S. Rangan: 'Generalized approximate message passing for estimation with random linear mixing'. In*Proceedings of the IEEE International Symposium on Information Theory Proceedings. (ISIT 2011)*. July 2011: pp. 2168–2172 (cit. on pp. 9, 33).

[121] S. Rangan, P. Schniter, and A. Fletcher: 'On the convergence of approximate message passing with arbitrary matrices'. In*Proceedings of the IEEE International Symposium on Information Theory. (ISIT 2014)*. June 2014: pp. 236–240 (cit. on p. 31).

[122] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher: 'Fixed points of generalized approximate message passing with arbitrary matrices'. In*Proceedings of the IEEE International Symposium on Information Theory Proceedings. (ISIT 2013)*. July 2013: pp. 664–668 (cit. on p. 33).

[123] G. Reeves and M. Gastpar: 'A note on optimal support recovery in compressed sensing'. In*Proceedings of the Forty-Third Asilomar Conference on Signals, Systems and Computers, (Asilomar 2009)*. Nov. 2009: pp. 1576–1580 (cit. on p. 14).

[124] G. Reeves and M. Gastpar: 'The sampling rate-distortion tradeoff for sparsity pattern recovery in compressed sensing'. In*IEEE Trans. on Inform. Theory* (2012), vol. 58(5): pp. 3065–3092 (cit. on p. 14).

[125] G. Reeves and M.C. Gastpar: 'Approximate sparsity pattern recovery: Information-theoretic lower bounds'. In*IEEE Trans. on Inform. Theory* (June 2013), vol. 59(6): pp. 3451–3465 (cit. on pp. 14, 36).

[126] A. Rényi: 'On the dimension and entropy of probability distributions'. In*Acta Mathematica Academiae Scientiarum Hungarica* (1959), vol. 10(1-2): pp. 193–215 (cit. on pp. 28, 29).

[127] A. Rényi: *Probability Theory*. Dover Books on Mathematics Series. Dover Publications, 2007 (cit. on p. 28).

[128] A. Rényi: 'Probability theory'. In*North-Holland Ser Appl Math Mech* (1970), vol. (cit. on pp. 28, 29).

[129] T.J. Richardson and R.L. Urbanke: *Modern Coding Theory*. New York, NY, USA: Cambridge University Press, 2008 (cit. on pp. 6, 8, 12, 17, 19, 37, 62).

[130] T.J. Richardson and R.L. Urbanke: 'The capacity of low-density parity-check codes under message-passing decoding'. In*IEEE Trans. on Inform. Theory* (Feb. 2001), vol. 47(2): pp. 599–618 (cit. on p. 9).

[131] C. Sae-Young: 'On the construction of some capacity-approaching coding schemes'. PhD thesis. Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2000 (cit. on p. 57).

[132] S. Sarvotham, D. Baron, and R.G. Baraniuk: 'Sudocodes - Fast measurement and reconstruction of sparse signals'. In*Proceedings of the IEEE International Symposium on Information Theory. (ISIT 2006)*. July 2006: pp. 2804–2808 (cit. on pp. 37, 38, 116, 118, 119, 122, 123).

[133] S.A. Savari and R.G. Gallager: 'Arithmetic coding for memoryless cost channels'. In*Proceedings of the Data Compression Conference. (DCC 1992)*. Mar. 1992: pp. 92–101 (cit. on p. 12).

[134] J. Scarlett and V. Cevher: *Phase transitions in group testing*. Tech. rep. 2015 (cit. on pp. 15–17).

[135] P. Schniter: 'Turbo reconstruction of structured sparse signals'. In*Proceedings of the 44th Annual Conference on Information Sciences and Systems. (CISS 2010)*. Mar. 2010: pp. 1–6 (cit. on pp. 34, 97, 119).

[136] P. Schniter and S. Rangan: 'Compressive phase retrieval via generalized approximate message passing'. In*IEEE Trans. on Signal Processing* (Feb. 2015), vol. 63(4): pp. 1043–1055 (cit. on p. 27).

[137] D.H. Schonberg: 'Practical distributed source coding and its application to the compression of encrypted data'. PhD thesis. EECS Department, University of California, Berkeley, July 2007 (cit. on p. 57).

[138] C. E. Shannon: 'Coding theorems for a discrete source with fidelity criterion'. In. Ed. by R. E. Machol. McGraw-Hill, 1960: pp. 93–126 (cit. on p. 11).

[139] C.E. Shannon: 'A mathematical theory of communication'. In*SIGMOBILE Mob. Comput. Commun. Rev.* (1 Jan. 2001), vol. 5: pp. 3–55 (cit. on pp. 5, 11).

[140] N. Shental, A. Amir, and O. Zuk: 'Identification of rare alleles and their carriers using compressed sensing'. In*Nucleic acids research* (2010), vol. 38(19): e179–e179 (cit. on p. 15).

[141] H. Shi, H. Zhang, G. Li, and X Wang: 'Stable embedding of Grassmann manifold via Gaussian random matrices'. In*IEEE Trans. on Inform. Theory* (May 2015), vol. 61(5): pp. 2924–2941 (cit. on p. 99).

[142] A. Shokrollahi: 'LDPC Codes: An introduction'. In*Coding, Cryptography and Combinatorics*. Vol. 23. Progress in Computer Science and Applied Logic Ser. Springer-Verlag New York, LLC, June 2004: pp. 85–110 (cit. on pp. 12, 19, 41).

[143] M. Sipser and D.A. Spielman: 'Expander codes'. In*IEEE Trans. on Inform. Theory* (Nov. 1996), vol. 42(6): pp. 1710–1722 (cit. on p. 37).

[144] M. Smieja and J. Tabor: 'Entropy of the mixture of dources and entropy dimension'. In*IEEE Trans. on Inform. Theory* (May 2012), vol. 58(5): pp. 2719–2728 (cit. on p. 28).

[145] M. Sobel and P.A. Groll: 'Group testing to eliminate efficiently all defectives in a binomial sample'. In*Bell System Technical Journal* (1959), vol. 38(5): pp. 1179–1252 (cit. on p. 15).

[146] S. Som and P. Schniter: 'Compressive imaging using approximate message passing and a Markov-tree prior'. In*IEEE Trans. on Signal Processing* (July 2012), vol. 60(7): pp. 3439–3448 (cit. on p. 27).

[147] J.H. Sorensen, P. Popovski, and J. Ostergaard: 'Design and analysis of LT codes with decreasing ripple size'. In*IEEE Trans. on Communications* (2012), vol. 60(11): pp. 3191–3197 (cit. on p. 13).

[148] A. Tajer, R. Castro, and X. Wang: 'Adaptive spectrum sensing for agile cognitive radios'. In*Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2010)*. Mar. 2010: pp. 2966–2969 (cit. on p. 15).

[149] J. Tan, Y. Ma, and D. Baron: 'Compressive imaging via approximate message passing with image denoising'. In*IEEE Trans. on Signal Processing* (Apr. 2015), vol. 63(8): pp. 2085–2092 (cit. on p. 27).

[150] M.E. Tipping: 'Regression shrinkage and selection via the LASSO'. In*Journal Royal. Statist. Soc* (1996), vol. 5(1): pp. 267–288 (cit. on p. 32).

[151] J.A. Tropp and A.C. Gilbert: 'Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit'. In*IEEE Trans. on Inform. Theory* (Dec. 2007), vol. 53(12): pp. 4655–4666 (cit. on pp. 31, 68).

[152] X.M. Tu, E. Litvak, and M. Pagano: 'On the informativeness and accuracy of pooled testing in estimating prevalence of a rare disease: application to HIV screening'. In*Biometrika* (1995), vol. 82(2): pp. 287–297 (cit. on p. 15).

[153] J.P. Vila and P. Schniter: 'Expectation-maximization Bernoulli-Gaussian approximate message passing'. In*Proceedings of the Forty Fifth Asilomar Conference on Signals, Systems and Computers. (ASILOMAR 2011)*. Nov. 2011: pp. 799–803 (cit. on p. 128).

[154] J.P. Vila and P. Schniter: 'Expectation-maximization Gaussian-mixture approximate message passing'. In*IEEE Trans. on Signal Processing* (Oct. 2013), vol. 61(19): pp. 4658–4672 (cit. on p. 34).

[155] J.P. Vila, P. Schniter, and J. Meola: 'Hyperspectral unmixing via turbo bilinear approximate message passing'. In*CoRR* (2015), vol. abs/1502.06435 (cit. on p. 27).

[156] T. Wadayama: 'An analysis on non-Adaptive group testing based on sparse pooling graphs'. In*arXiv preprint arXiv:1301.7519* (2013), vol. (cit. on p. 15).

[157] T. Wadayama: 'An analysis on non-adaptive group testing based on sparse pooling graphs'. In*Proceedings of the IEEE International Symposium onInformation Theory Proceedings (ISIT 2013)*. July 2013: pp. 2681–2685 (cit. on pp. 15, 51).

[158] M.J. Wainwright: 'Sharp thresholds for high-dimensional and noisy sparsity recovery using l_1 -Constrained Quadratic Programming (Lasso)'. In*IEEE Trans. on Inform. Theory* (May 2009), vol. 55(5): pp. 2183–2202 (cit. on p. 32).

[159] M.J. Wainwright and E. Maneva: 'Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes'. In*Proceedings of the International Symposium on Information Theory. (ISIT 2005)*. Sept. 2005: pp. 1493–1497 (cit. on p. 11).

[160] J. Wolf: 'Born again group testing: Multiaccess communications'. In*IEEE Trans. on Inform. Theory* (Mar. 1985), vol. 31(2): pp. 185–191 (cit. on p. 15).

[161] Y. Wu: 'Shannon theory for compressed sensing'. PhD thesis. Princeton University, 2011 (cit. on pp. 3, 27–29).

[162] Y Wu and S. Verdu: 'Optimal phase transitions in compressed sensing'. In*IEEE Trans. on Inform. Theory* (Oct. 2012), vol. 58(10): pp. 6241–6263 (cit. on pp. 5, 28, 29).

[163] Y Wu and S. Verdu: 'Renyi information dimension: Fundamental limits of almost lossless analog compression'. In*IEEE Trans. on Inform. Theory* (Aug. 2010), vol. 56(8): pp. 3721–3748 (cit. on p. 28).

[164] J.S. Yedidia, W.T. Freeman, and Y. Weiss: 'Constructing free-energy approximations and generalized belief propagation algorithms'. In*IEEE Trans. on Inform. Theory* (July 2005), vol. 51(7): pp. 2282–2312 (cit. on p. 9).

[165] S.A. Zenios and L.M. Wein: 'Pooled testing for HIV prevalence estimation: exploiting the dilution effect'. In*Statistics in medicine* (1998), vol. 17(13): pp. 1447–1467 (cit. on p. 15).

[166] F. Zhang and H.D. Pfister: 'Verification decoding of high-rate LDPC codes with applications in compressed sensing'. In*IEEE Trans. on Inform. Theory* (Aug. 2012), vol. 58(8): pp. 5042–5058 (cit. on pp. 35, 115, 116, 119, 123).

[167] F. Zhang and H.D. Pflster: 'List-message passing achieves capacity on the q-ary symmetric channel for large q'. In*IEEE Global Telecommunications Conference. (GLOBECOM2007)*. Nov. 2007: pp. 283–287 (cit. on p. 124).

[168] Q. Zhang, A.B.J. Kokkeler, and G.J.M. Smit: 'An efficient multi-resolution spectrum sensing method for cognitive radio'. In*Third International Conference on Communications and Networking in China. (ChinaCom 2008)*. Aug. 2008: pp. 1226–1229 (cit. on p. 27).

[169] W. Zhang, A.K. Sadek, Shen C., and S.J. Shellhammer: 'Adaptive spectrum sensing'. In*Information Theory and Applications Workshop*. Feb. 2010: pp. 1–7 (cit. on p. 15).

[170] J. Ziv and A. Lempel: 'A universal algorithm for sequential data compression'. In*IEEE Trans. on Inform. Theory* (May 1977), vol. 23(3): pp. 337–343 (cit. on pp. 3, 12).

[171] J. Ziv and A. Lempel: 'Compression of individual sequences via variable-rate coding'. In*IEEE Trans. on Inform. Theory* (Sept. 1978), vol. 24(5): pp. 530–536 (cit. on pp. 3, 12).