Imperial College London

Department of Electrical and Electronic Engineering

Hierarchical Bayesian Models for Sparse Signal Recovery and Sampling

Evripidis Karseras

July 2015

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College London and the Diploma of Imperial College London

Abstract

This thesis builds upon the problem of sparse signal recovery from the Bayesian standpoint. The advantages of employing Bayesian models are underscored, with the most important being the ease at which a model can be expanded or altered; leading to a fresh class of algorithms. The thesis fills out several gaps between sparse recovery algorithms and sparse Bayesian models; firstly the lack of global performance guarantees for the latter and secondly what the signifying differences are between the two. These questions are answered by providing; a refined theoretical analysis and a new class of algorithms that combines the benefits from classic recovery algorithms and sparse Bayesian modelling. The said Bayesian techniques find application in *tracking* dynamic sparse signals, something impossible under the Kalman filter approach.

Another innovation of this thesis are Bayesian models for signals whose components are known *a priori* to exhibit a certain statistical trend. These situations require that the model enforces a given statistical bias on the solutions. Existing Bayesian models can cope with this input, but the algorithms to carry out the task are computationally expensive. Several ways are proposed to remedy the associated problems while still attaining some form of optimality. The proposed framework finds application in multipath channel estimation with some very promising results.

Not far from the same area lies that of Approximate Message Passing. This includes extremely low-complexity algorithms for sparse recovery with a powerful analysis framework. Some results are derived, regarding the differences between these approximate methods and the aforementioned models. This can be seen as preliminary work for future research.

Finally, the thesis presents a hardware implementation of a wideband spectrum analyser based on sparse recovery methods. The hardware consists of a Field-Programmable Gate Array coupled with an Analogue to Digital Converter. Some critical results are drawn, regarding the gains and viability of such methods.

I, Evripidis Karseras, hereby declare that the entirety of the work presented in this thesis was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the bibliography at the end of this document.

Acknowledgements

Firstly I would like to express my gratitude to my supervisor Dr. Wei Dai for his genuine support, accurate guidance and for taking me in his research group under some adverse circumstances. These words extend equally to my secondary supervisor Professor Kin K. Leung with whom I had the chance to have some very fruitful discussions during the course of the SMARTEN ITN project.

It would be a great omission not to thank Dr. Miguel Rodrigues of the Department of Electronic and Electrical Engineering, University College London and Professor Pier Luigi Dragotti for accepting my invitation to serve as my external and internal examiners respectively.

I would also like to acknowledge the European Commission for funding SmartEN (Grant No. 238726) under the Marie Curie ITN FP7 program, as the research work presented here is partially supported by this program. The opinions expressed in this dissertation do not necessarily reflect those of the sponsors.

Contents

List of Figures

List of Tables

List of Algorithms

List of Mathematical Notation

W	Upper-case letters are used to denote constants.
t	Lower-case letters are used to denote scalar variables.
$f(t)$	A function f with argument t. Used in the text to denote a continuous-time signal.
$f[n]$	A function f with argument n. Brackets are used to indicate that the argument is a discrete variable. Used in the text to denote a discrete-time signal.
$\Re\{x\}, \Im\{x\}$	Denote the real and imaginary parts of x respectively.
$\mathcal{F}(\omega)$	Calligraphic upper-case letters used to indicate a function with a special meaning to the context. Usually used to denote a transform or a cost-function.
$\mathcal{O}$,o	Calligraphic upper-case and lower-case Greek letter omicron, reserved to denote the big-O and little-o asymptotic notation.
$\boldsymbol{f}$	Bold-face lower-case letters are used to denote vectors.
$\boldsymbol{f}_{\mathcal{I}}$	Used to denote a vector formed by the entries of vector $\boldsymbol{f}$ indexed by the elements of set $\boldsymbol{\mathcal{I}}$.
$\boldsymbol{b}_k$	Used to indicate a vector indexed by k in a set. Also used to indicate the k^{th} column vector of the corresponding matrix $\boldsymbol{B}$.
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle$	The inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.
$\boldsymbol{x}_{t_1:t_2}$	Denotes the values of vector $\boldsymbol{x}$ taken from time instants t_1 through t_2.
x_i	Denotes the i^{th} entry of the corresponding vector $\boldsymbol{x}$.
x_b^a	Subscripts and superscripts are used to describe a quantity x (scalar or vector) based on the context. The values of a, b can indicate time, iteration or context. In case a power index is used, a vector entry index or for the sake of clarity then the notation is abused slightly.
$\|\boldsymbol{a}\|_p$	The l_p norm of vector $\boldsymbol{a}$ with $1 \leq p < +\infty$.
$\|\boldsymbol{a}\|_0$	Counts the non-zero entries of vector $\boldsymbol{a}$.
A_{ij}	Used to denote a single entry of the corresponding matrix at row i and column j.
$\boldsymbol{A}$	Bold-face upper-case letters are used to denote matrices.

$[\boldsymbol{A}, \boldsymbol{B}]$	Denotes the concatenation of matrices $\boldsymbol{A}, \boldsymbol{B}$ into one single matrix. The dimensions are assumed legal so that the result also denotes a matrix.
$\boldsymbol{H}_{\mathcal{I}}$	Used to denote a matrix formed by the column vectors of matrix $\boldsymbol{H}$ indexed by the elements of set $\boldsymbol{\mathcal{I}}$.
$\boldsymbol{H}^T$	Denotes the transpose of a matrix.
$\boldsymbol{H}^H$	Denotes the Hermitian transpose of a matrix.
$\boldsymbol{I}$	The identity matrix the dimensions of which are assumed to be correct based on the context.
$\text{diag}(\boldsymbol{x})$	A zero matrix but with the entries of $\boldsymbol{x}$ on its main diagonal.
$\boldsymbol{\mathcal{G}}$	Calligraphic upper-case letters are used to denote sets.
$\boldsymbol{\mathcal{G}}^t$	Super-scripts on sets are used to denote the state of a set at iteration t.
$\mathbb{R}$	The set of real numbers.
$\mathbb{C}$	The set of complex numbers.
$\mathbb{R}^n$	The n-dimensional real space.
$\mathbb{R}^+$	The set of positive real numbers.
$\emptyset$	The empty set.
$\mathcal{A} - \mathcal{B}$	Denotes the set difference, $\{x : x \in \mathcal{A}, x \notin \mathcal{B}\}$.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	The Normal distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$.
$\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	The circularly-symmetric Complex Normal distribution with mean μ and variance $\boldsymbol{\Sigma}$.
$\text{Gamma}(a, b)$	The Gamma distribution with shape and scale parameters a, b respectively.
$\text{St}(a, b)$	The non-standardised zero-mean Student-t distribution with parameters a, b respectively.
$\Gamma(x)$	The Gamma function.
$\langle X \rangle$	The expected value of random variable $\text{E}(X)$. The two notations are used interchangeably.

Chapter 1

Introduction

The tragic poet *Euripides* (480 – 406 BC) mentioned in his works; "To be succinct, is to be wise" and later on, the polymath *Aristotle* (384 – 322 BC) stated; "Among equal demonstrations, the one which is derived from fewer postulates or hypotheses is superior". Equivalently, *The Principle of Parsimony* devises a philosophical framework put forward originally by an English friar, William of Ockham (AD 1287 – 1347); states in a nutshell that *the fewer the better*. From what has been known as Ockam's razor to the *"K.I.S.S" (Keep It Simple, Stupid)* motto of the United States' Navy in the 1960s; it always seems that primarily nature and secondly the human reason which is part of the latter are amenable and attracted to simple and plain explanations.

This simple principle about simplicity itself has found refuge in many applications of physics, chemistry, biology and others, but also in the field of probability theory which is akin to the subject of this thesis. Studies about it have been documented through the years at first empirically, then philosophically and also mathematically by Solomonoff's theory of inductive inference. This thesis is also based on this attribute, i.e., that the environment in which a process takes place is governed by a probability distribution which can be computable or hopefully approximated.

In their journey from classic signal processing and into the not-so-long but, definitely plentiful era of digital signal processing, engineers have always been troubled with *samples*. Quite common is the question of how many samples are required and the answer is subjected to constraints such as digital memory, communication bandwidth, power consumption and more. In the last decade, Shannon has met Ockham in the works of Emmanuel Candès, David Donoho and Terrence Tao to what is known in the community as *Compressed Sensing* and more generally in recent years as *Sparse Processing*. Simply put, if a signal can be explained simply in some domain then *not many* samples are required for its representation. Actually, far less than what Shannon's theorem suggests.

1.1 Sparse Signals and Bayesian Learning

This thesis comes after Compressed Sensing has been studied and well-accepted by the engineering field and ventures in the area that lies between Bayesian models and sparse signal recovery. Bayesian methods provide useful intuition and scalability. Usually a probabilistic model is constructed and then this model is trained using available data for the purpose of prediction or extrapolation. Another close relative is regression where usually the regression variables are taken to be the ones to be *inferred* from the dataset. The quality of the solution is amenable to the model mismatch, i.e., how the assumed model differs from the actual natural process that generated the data, but also the *inference algorithm* used to carry out what basically is an application of the Bayes theorem. In the particular case of sparse signal recovery the framework of *Sparse Bayesian Learning* (SBL) encompasses models of a specific structure that promotes sparsity in the inferred solutions.

Despite the existence of several flavours of the SBL shortly after the appearance of Compressed Sensing, the discussion about the performance guarantees of such models is limited and narrow, focusing on local guarantees and scenarios. Performance guarantees complement a specific algorithm or principle for solving a class of problems and provides the answer to the all-important question, which in our case, is whether the number of samples is sufficient to recover a sparse signal. The thesis puts one step forward into asking and answering this question for SBL; "Is this number of signals sufficient to *model* a sparse signal ?". In the main body of this text it is shown that SBL has many close ties with traditional Compressed Sensing algorithms and these connections are quantified exactly. Based on this innovation, it is then possible not only to derive performance guarantees on SBL but also to improve it without sacrificing the generality of the model.

1.1.1 Dynamic Sparse Signals

Quite close to the area of Bayesian estimation is that of Linear Dynamical Systems. In more loose terms this can be seen as sequential Bayesian estimation where the required statistics (for example the mean and the variance) of a predetermined model are updated upon each new sample. More specifically, this is an area in which the Kalman filter has been considered to be the workhorse in uncountable applications for many years. With the advent and gained popularity of sparse signal recovery a gap is generated since there is enough knowledge to perfectly reconstruct a sparse signal but not to *track* it in a temporal manner, since the Kalman filter cannot support a sparse signal model.

In one of the chapters of this thesis this problem is exemplified and an elegant way for its solution is presented. The solution is based on the SBL framework and

the innovations on the latter. The existing techniques are also discussed in depth and it is argued why the proposed methods are indeed novel. What has been discovered is that the Kalman filter system model can be expanded (rather than supplemented by external modifications which was the case with other approaches) with ideas from SBL at very little computational expense while at the same time taking advantage of the temporal correlation between the measurements to achieve lower reconstruction error. Probably the most interesting of the empirical results is that of reconstructing the daily Ozone measurements from an experiment performed on a shuttle of the United States' National Aeronautics and Space Administration (NASA). The proposed technique was employed in order to recover a damaged dataset due to a sensory malfunction.

1.2 Informative Sparse Bayesian Models

Not diverting far from the same research ideas this part of the thesis deals with a rather overlooked part of SBL. In most cases the sparse signal to be recovered is assumed to be completely random with the only prior knowledge that it is sparse. This aids in adopting a suitable model but the cases where *additional statistical bias* is available are not studied adequately in the bibliography. Such prior biases might come either from expert opinions or pure empirical knowledge and using such knowledge can lead to improved performance of the model by minimising mismatch with the actual system.

In SBL, such a mechanism exists but it is rarely exploited mostly due to the simple fact that sparse recovery is the goal and the so-called *uninformative* SBL is used. To be more dubious; the fact that the structure of the model is taken to be *uninformative* has allowed fast inference algorithms to be constructed. In the thesis it is argued that this *is not* the case when the model is subjected to a bias and the whole "fast" machinery fails to perform. A careful study is performed and several results are drawn that allow a statistical bias to be used in SBL but allowing at the same time efficient inference algorithms to be drawn. The resulting algorithms are optimal with respect to a certain metric.

1.2.1 Multipath Channel Estimation

Figure 1.1 shows the results from a private wireless over-the-air transmissions experiment (more details in the main body of the text). The mean of the estimated channel coefficients is drawn with red dots while the lines above and below are the error bars for one standard deviation. The power-delay profile of this channel is met in numerous cases and it is better known as a *multipath fading channel*. When

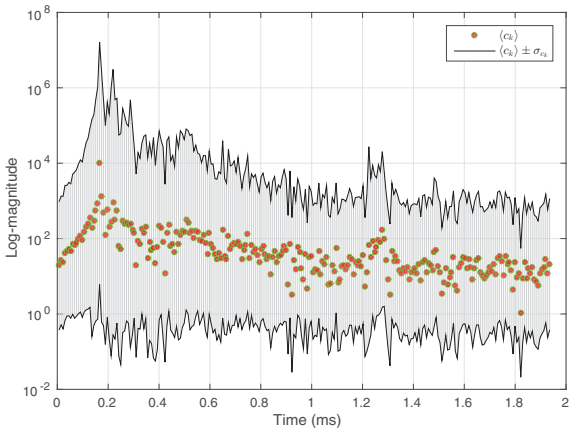


Figure 1.1: *Empirical channel response measurements.*

viewed under the sparsity lens this class of signals can be described as compressible, i.e., the number of significant components is small compared to the whole. Indeed, from the empirical tests one observes that the strongest channel coefficients are located closer to the receiver with an amplitude several classes of magnitude larger than the rest. To be more pedantic it is evident that they follow an *exponential* decay law.

Based on these two ingredients - sparsity and exponential trend of the signal - the *informative* SBL is employed to improve the channel equalisation performance of pilot-assisted Orthogonal Frequency Division Multiplex (OFDM) systems. The *uninformative* SBL is able to model successfully a sparse signal but when it comes to using the fact that there is a trend in the components the *informative* SBL has to be employed so as to impose a certain preference over the channel coefficients that are closer to the receiver. Moreover, in the OFDM context it is shown that a reduction in the pilot symbol overhead is possible in addition to the Bit Error Rate (BER) improvement.

1.3 The Approximate Message Passing Framework

Close to the SBL framework is that of message passing and its approximations (AMP). By adopting a different way of thinking, Bayesian inference is achieved differently with results that ultimately end up being quite different and have surprisingly sparked the starting point for an exciting research area. In the message passing framework, the distributions that relate the individual components of a model are assumed to be "messages" exchanged between them. It was proven in the bibliography by David Donoho, Andrea Montanari *et al.* [32, 33, 7] that an *approximation* to this scheme results in extremely efficient inference algorithms for sparse recovery. As opposed to the SBL, these algorithms are free from burdensome computations

such as matrix inversions. Despite its *simplicity*, in a tremendously fresh framework it is proven that *asymptotically*, i.e., for large systems, that the AMP achieves the exact theoretical recovery bounds as those suggested by compressed sensing.

In what can be considered as the starting point for future work, in the final parts of this thesis a fusion is attempted between the SBL estimators and the AMP paradigm. In the AMP analysis a central role is played by the signal estimator for scalar quantities. Also a key element is that of the *worst-case distribution* for an estimator, which is that signal distribution which maximises distortion for a given estimator. The scalar estimator and the worst case distribution are derived for the SBL model and a comparison is attempted with the aim to understand where the two techniques differ. Some surprising results are derived.

1.4 Wideband Spectrum Sensing: Hardware for Compressed Sensing

One of the drawbacks of Compressed Sensing from the very beginning has been the lack of appropriate hardware. Most of the digital signal processing techniques that are applied in many cases ranging from compact music playback devices to highly sophisticated defence equipment rely on the fact that signals are to be sampled regularly and at a rate that obeys Shannon's sampling theorem. Compressed Sensing on the other hand imposes some very specific constraints on the sampling regime to be used, something which is proven difficult to implement in practice. Applying compressed sensing on traditional hardware is a delicate subject.

This is the subject of the final part of this thesis. Compressed sensing is implemented on a Field Programmable Gate Array (FPGA) device alongside a traditional Analogue-to-Digital Converter. The FPGA runs a full processing core developed in-house that implements the AMP algorithm. The prototype to the author's knowledge is known to be the first of its kind, in the confines of the United Kingdom and probably the only one to have implemented the specific techniques described in the relevant chapter. The conclusions are interesting, to say the least, and most of all unprecedented and demystifying. The apparatus deals with wideband spectrum sensing as the application with the aim of handling a bandwidth of $0.4 - 1.6$ GHz instantaneously. A series of technical issues is documented and several algorithmic aspects are studied empirically. These provide a heap of possible future work directions.

1.5 Thesis Roadmap

In the diagram of Figure 1.6 (last page of this chapter), a roadmap for the contents of the thesis is depicted. The blue colour represents the greater areas with which the thesis transacts. In grey color are the specific theoretical thrusts that are followed. These lead to the red boxes which represent the main theoretical innovations of the thesis. Finally in orange colour are the main practical applications of the innovations.

A short description of the individual chapters is:

- In **Chapter 2** takes place a summary of the milestones in the area of sparse signal recovery. This covers some of the most important theoretical results in this field that are relevant with the thesis.

- In **Chapter 3** follows a short summary of some of the hardware architectures that have been proposed in the bibliography for compressed sensing.

- **Chapter 4** proceeds in describing the SBL framework and it is shown how some efficient sparse recovery algorithms can be formulated in a completely Bayesian setting.

- In **Chapter 5** the it is demonstrated how the *uninformative* SBL can be extended and improved. The presented theoretical analysis also leads to the relevant global performance guarantees.

- In **Chapter 6** the Kalman filter system model is extended with the SBL model in order to successfully track dynamic sparse signals. One of the considered scenarios is that of the missing data problem in an Ozone measurement experiment by NASA.

- Follows in **Chapter 7** a study of the *informative* SBL and an analysis which allows for efficient inference algorithms to be constructed.

- In **Chapter 8** the fast *informative* SBL is taken into consideration for multi-path channel estimation in pilot-assisted OFDM systems.

- **Chapter 9** presents a summary of the AMP analysis framework and some results are presented when this analysis is applied on the SBL model.

- **Chapter 10** A complete FPGA hardware implementation of the AMP algorithm for wideband spectrum sensing is presented. The results are known to be seminal.

1.5.1 Thesis Contributions and Innovations

The contributions of this thesis are outlined below:

- *Contribution 1*: The derivation of the exact connection between Sparse Bayesian Learning and greedy pursuit algorithms for sparse signal recovery. This is explained in detail in Chapter 5. This contribution extends previous work of other researchers regarding the connection of sparse Bayesian models and basis selection. This relationship manages to make the situation a lot more clear on how sparsity is realised during the inference phase in such models.

- *Contribution 2*: Taking advantage of Contribution 1, it is explained in Chapter 5 how it is possible to derive inference algorithms based on greedy pursuit algorithms with sophisticated recovery schemes. Indeed by analysing the traditional algorithm used in Sparse Bayesian Learning it is shown that its sufficient condition for exact sparse signal recovery can be improved by redesigning some of its features. It is also demonstrated how the resulting improved algorithms can have provable performance guarantees.

- *Contribution 3*: The probabilistic model behind Sparse Bayesian Learning is employed to solve the problem of tracking dynamic sparse signals. It is demonstrated in Chapter 6 how the said model can be incorporated in the Kalman filter to track a sparse signal. It is a fact that the traditional Kalman filter cannot accommodate sparse solutions. Results from Contribution 2 are also used to further improve tracking. Empirical results are presented for both synthetic and real-world scenarios.

- *Contribution 4*: A major drawback of the aforementioned model is pointed out in Chapter 7. Basically it is shown that when certain assumptions are altered then efficient algorithms for inference cannot be used. This contribution is towards establishing a set of alternations that make efficient inference possible.

- *Contribution 5*: The said changes in the assumptions regard cases where a certain prior statistical bias is available for the sparse components that are to be inferred. A special case presented in Chapter 8 is that of multipath channel estimation in OFDM systems. The priorly available bias is that the channel coefficients are likely to follow an exponential decay trend while moving away from the receiver and for the channel duration. It is shown that it is possible to use such information to achieve lower Bit Error Rates and smaller pilot symbol numbers.

- *Contribution 6*: In Chapter 10 a prototype wideband spectrum analyser is presented that employs compressed sensing to recover a bandwidth of 0.4-1.6

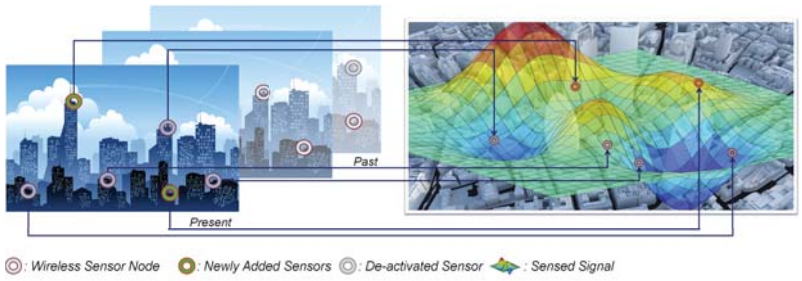
GHz directly. The recovery algorithm used is the Approximate Message Passing and the sampling technique is based on Discrete Random Sampling, i.e., a randomly clocked Analogue-to-Digital-Converter. The recovery algorithm is implemented on an FPGA and some results from field tests are presented.

- *Contribution 7*: This contribution is minor and can be seen as future work. In Chapter 9 an attempt is made to find the connections between the Approximate Message Passing inference algorithm and Sparse Bayesian Learning. The motivation behind this is that both schemes involve approximate inference for a model that involves similar prior distributions. It was envisaged that any possible connections can be used to bridge the two. The first steps for this are shown in Chapter 9 with a comparison of the scalar estimators of the two and their corresponding worst-case mean squared error performance.

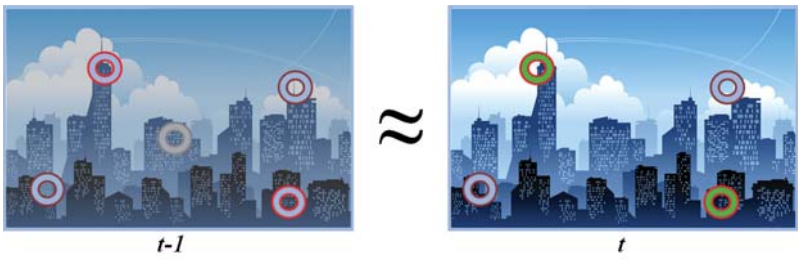
1.6 The SMARTEN Project

The SMARTEN project kicked-off in 2010 and was funded by the European Commission under the FP7 People actions. The ultimate goal of this project was to address the issues of managing the environment in which the modern human lives and acts in a smart manner by employing novel digital signal processing techniques oriented towards miniaturised wireless sensor networks. The niche aspect of this project was its multidisciplinary nature, combining signal processing and civil engineering in order to achieve a *sustainable* environment that takes into account the ageing civil infrastructure, non-destructive evaluation, the ever-changing city microclimates and the specific constraints placed on the wireless sensing part (sensing elements, power consumption, communications bandwidth, data fusion, antenna design).

The author was involved in the aforementioned project via a collaboration between DFL Systems Ltd. and Imperial College London. DFL Systems is a Research and Design company based in the United Kingdom specialising in the application of signal processing in electronic systems providing expert advise to various blue chip companies. The author was involved as an *Early Stage Researcher* working in *Work Package 2* of the SMARTEN project dealing with Sensor Signal Processing. Figure 1.2 depicts an abstraction of a relevant wireless sensor network over an urban area for the purpose of sampling an underlying, time-varying signal. Over time, not only the signal varies but the sensor network itself, being affected by external factors (for example harsh environments), vandalisms, poor reception, battery status or simply the need to add or remove sensors to alter the coverage area. The techniques that have been established theoretically in this thesis find application in such scenarios.



(a) *Sensor network over an urban area.*

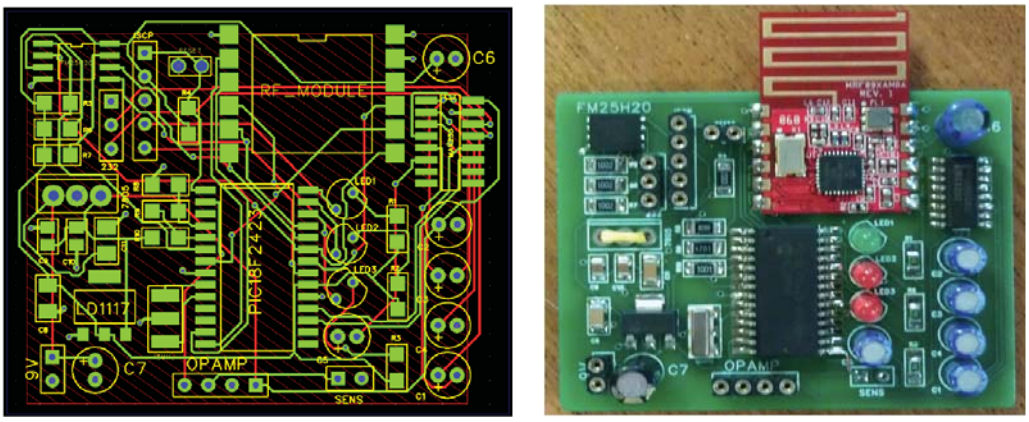


(b) *Temporal changes affecting the network.*

Figure 1.2: *(a) An abstraction of a wireless sensor network in an urban area. (b) Over time the sensed signal changes but the network changes as well, i.e., sensors brake, batteries deplete, new ones are added etc.*

During the course of the project and for experimental purposes a miniaturised wireless sensor network has been developed in the premises of DFL Systems. The sensor itself including analogue design, firmware development, circuit board design and manufacturing was performed by the author as a part of his training. The end-product can be seen in Figure 1.3.

This sensor network was used during the secondment of another Early Stage Researcher at DFL Systems for the empirical study the effects of humidity on the transceiver's received signal strength. Four nodes have been used with one of them acting as a base station. A minimal routing protocol was developed also on site with the collaboration of the seconded researcher. A week's worth of measurements had been acquired and each node reported at each transmission which was the parent node. It was then possible to relate the sensed humidity values with the changes in the parent nodes. The nodes where positioned in such a way so as to allow for the nodes to select a different parent note based on the received signal strength which depends on the distance between the nodes and the sensor placement. The measurements were being uploaded real-time to an on-line feed host. The status of the gateway node was also being checked remotely via a command-line secure shell connection. The results of this study have been documented in an internal extensive



(a) *Printed circuit board routing.* **(b)** *Assembled sensor.*

Figure 1.3: *One of the wireless sensor nodes that were developed.*

report. The sensor network was deployed also on site at DFL Systems. A map and a mounted sensor can be seen in Figure 1.4.

(a) **(b)**

Figure 1.4: *(a) An overview of the deployed nodes. (b) A sensor mounted on a drainpipe under a steel shed.*

For the purpose of the SMARTEN project, the author was seconded at the Civil Engineering department of the University of Surrey. The hosting research team was investigating ways to analyse the corrosion on steel reinforcing bars in concrete structures. The experiments took place in a controlled environment in an accelerated test procedure, where a high current was induced on the steel bars that have been enclosed in hardened concrete. The amperage and application time was passed on to a model that related these two parameters to a number of actual years of the structure being exposed to corrosion by environmental factors. The bars where broken out of the concrete and then scanned by a high-precision three dimensional scanner and the un-wrapped cylindrical images where sent to a personal computer for analysis in raw format. Such an example can be seen in Figure 1.5.

The images where then subjected to wavelet analysis in order to automatically recover the corrosion pit severity and build a dataset with pit severity, age of steel bar

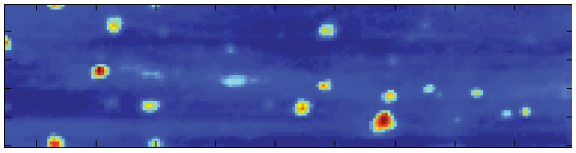


Figure 1.5: *Corrosion pits on a 3D scanned steel reinforcement bar (courtesy of Dr. Leticia Llano Trueba).*

and other parameter. The author was involved in aiding the host team with relevant signal processing techniques for multi-resolution signal analysis and implementation. There have also been attempts to apply prediction techniques so as to extrapolate results for a number of years with the ultimate goal to save experimental time. Results have also been documented in an internal report.

1.7 Academic Papers and Project Reports

During the course of building this thesis, several papers have been produced. To the author's knowledge it has always been the case that the presented ideas and statements to always be original and documented to their full extent. Moreover, there has been a significant amount of effort to always position the proposed work in the suitable context by achieving the appropriate balance between breadth and width.

At the time of writing the following journal papers are in preparation:

1. Evripidis Karseras and Wei Dai *"Improved Inference in Sparse Bayesian Learning for Dynamic Sparse Signal Tracking"*

2. Evripidis Karseras and Wei Dai *"Informative Sparse Bayesian Learning for Channel Estimation With Prior Statistical Input"*.

The following papers have been accepted for presentation in peer-reviewed conferences:

1. Evripidis Karseras, Kin Leung, and Wei Dai *"Tracking Dynamic Sparse Signals using Hierarchical Bayesian Kalman Filters"*, International Conference on Acoustics, Speech, and Signal Processing (2013)

2. Evripidis Karseras, Kin Leung and Wei Dai *"Tracking Dynamic Sparse Signals with Kalman Filters: Framework and Improved Inference"*, International Conference on Sampling Theory and Applications (2013)

3. Jason Filos, Evripidis Karseras, Wei Dai and Shulin Yan *"Tracking Dynamic Sparse Signals with Hierarchical Kalman Filters: A Case Study"*, Digital Signal Processing (2013)

4. Evripidis Karseras, Kin Leung, and Wei Dai *"Bayesian Compressed Sensing: Improving Inference "*, China Summit and International Conference on Signal and Information Processing (2013)

5. Evripidis Karseras, Kin Leung and Wei Dai *"Hierarchical Bayesian Kalman Filters for Wireless Sensor Networks "*, European Signal Processing Conference (2013)

6. Evripidis Karseras and Wei Dai *"A Fast Variational Approach for Bayesian Compressive Sensing with Informative Priors"*, International Conference on Acoustics, Speech, and Signal Processing (2014)

7. Evripidis Karseras, Wei Dai, Linglong Dai and Zhaocheng Wang *"Fast Variational Bayesian Learning for Channel Estimation with Prior Statistical Information"*, International Workshop on Signal Processing Advances in Wireless Communications (2015)

The following reports have been prepared as deliverables for projects that have been undertaken:

1. Khash-Erdene Jalsan and Evripidis Karseras *"Evaluating Humidity Effect on the received Signal Strength of a Radio Transceiver"*, Joint Secondment Report, DFL Systems Ltd. (2012)

2. Bo Han, Evripidis Karseras *"Investigation of Distortion on Audio Amplifiers and Ultrasonic Range Detection"*, Joint Secondment Report, DFL Systems Ltd. (2012)

3. Evripidis Karseras and Leticia Llano Trueba *"Sparse Representations for Predicting Corrosion Pit Depth of Reinforcing Bars in Concrete"*, Joint Secondment Report, Civil Engineering Department, University of Surrey (2013)

4. Evripidis Karseras *"D19-Research Report for the European Commission FP7 Project SMARTEN: Bayesian Models for Dynamic Sparse Signals - Tracking and Inference"*, Scientists in-charge: Dr. Wei Dai, Professor Kin K. Leung and Dr. Richard Orme (2013)

5. Evripidis Karseras, Wei Dai, and Kin Leung *"Dynamic Sparse Signal Models for Sustainable Monitoring of the Human Environment"*, SMARTEN ITN Final Conference (2013)

6. Wei Dai, Evripidis Karseras and Cong Ling *"CDE31737: Efficient Wideband Spectrum Surveillance for Situational Awareness - SWAPC Reduction via a Compressed Sensing Implementation"*, Final Report Prepared for The United Kingdom Ministry of Defence, DSTL (2014).

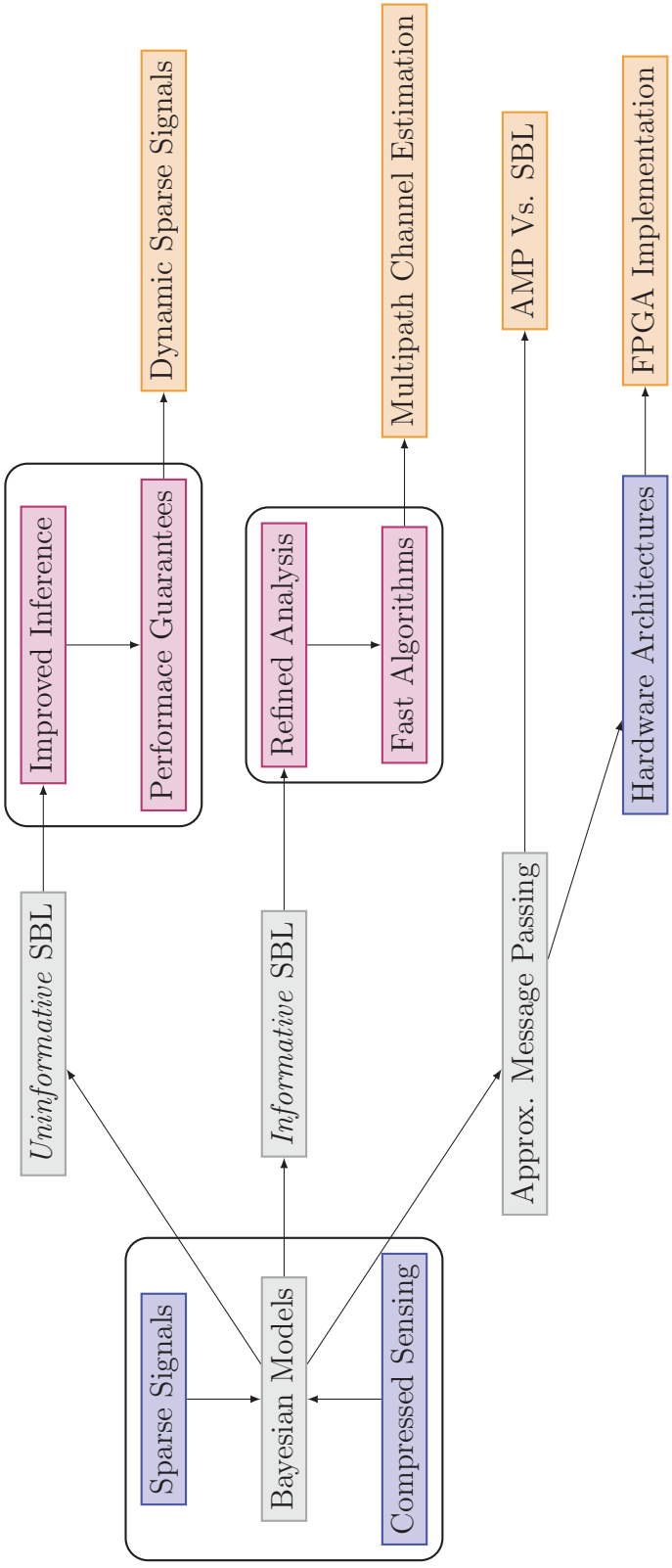


Figure 1.6: *Diagram visualising the progress of the research collocated in the thesis.*

Chapter 2

Sparse Signal Representations

In the digital signal processing realm, analogue signals are digitised so that certain mathematical tools can be applied and information can be extracted from the signal itself. For example voice signals are sampled and then the cepstral coefficients are estimated. Another example is that of communications' signals where an analogue signal is transmitted over a medium and then via a series of digital processing stages the transmitted data are recovered. In most of the cases in order for the transmitted *information* to be acquired, high sampling rates are required and also careful design and implementation of the analogue front-end.

The concept of *sparse signals* is that signals can be compactly represented in some domain different than their natural one. A simple sinusoid signal is spread out in the time domain but it is localised in the frequency domain. Results from information theory have shown that it is possible to efficiently sample and accurately reconstruct sparse signals. The *Compressive Sensing* framework accomplishes to capture all the necessary information in a signal without extraneous samples and also in a uniform manner since all acquired samples are deemed equally important. The significantly under-sampled signal can be accurately reconstructed by introducing the notion of *incoherence* (or randomness) in the sampling process. The trade-off for this gain is the additional computational requirements for signal reconstruction.

2.1 Classic Sampling Process

Usually the reason to measure a signal whether it is a voltage value at the output of a sensor or the price of a share in the stock market is to achieve a certain goal. A number of mathematical tools can be applied in order to extract useful information from the data points. Dual to this action is the process of suppressing unwanted information from the signal such as noise or even the isolation of specific structured parts of the signal.

A very good example from the area of digital signal processing is the suppression

of noise from a sampled electrical signal. A digital filter acts upon the samples of the signal altering its content according to some specification. It is recognised that by transforming the signal to the frequency domain it becomes possible to process the conveyed information in a much more efficient and meaningful way. In the cepstral analysis example; the cepstral coefficients can be used to easily to identify a spoken word.

By sampling, digitising and transforming a signal the unequivocal value of *sparsity* is recognised. As it happens for most of signals of interest the signal can be well approximated by a small number of components in some other domain. Not all frequencies are equally important in an audio signal, an image can be compactly represented in the wavelet domain or huge datasets might contain only a few classes of data. This fact on its own means great savings in the number of computations, the storage requirements, power consumption, communication overhead and many more.

2.1.1 Classic Sampling Theorem

Quoting Shannon from his original paper [83],

Theorem 1 (Shannon)**.** *If a function $f(t)$ contains no frequencies higher than W Hz, it is completely determined by giving its ordinates at a series of points spaced $1/2W$ seconds apart.*

Usually the function under study $f(t)$ is the associated continuous-time signal and its discrete-time counterpart; $f[n] = f(nT)$ a sequence of values taken at integer multiples of time interval T. The theorem provides a sufficient condition for the exact reconstruction of signal $f(t)$ from its samples $f[n]$, i.e., that the sampling period must obey $T \leq 1/2W$. The Fourier transform $\mathcal{F}(\omega)$ of $f(t)$ is assumed to exist so that $\mathcal{F}(\omega) = 0$ for $\omega \geq W$.

In his original paper Shannon discusses how $f(t)$ can be reconstructed from $f[n]$ via the *Whittaker-Shannon Interpolation Formula*,

$$f(t) = \sum_{n=-\infty}^{+\infty} f[n] \cdot \operatorname{sinc}\left(\frac{t - nT}{T}\right)$$

where $\operatorname{sinc}(x) = sin(\pi x)/\pi x$.

2.1.2 Aliasing and Interpolation Error

Focusing on Shannon's sampling theorem, two sources of degradation can exist. One of them is the well-known *aliasing* which is caused by sampling rates being lower

than what the theorem suggests. In short; lower parts of the sampled spectrum are added to the upper parts thus producing the spectrum of a signal different than the one being sampled. This is due to the fact that the uniform sampling of $x(t)$ at intervals nT results in the superposition of shifted copies of $\mathcal{F}(\omega)$ at integer multiples of $^1/_T$. The theorem suggests sampling rates so that this situation is avoided. This form of distortion can be avoided by proper filtering.

The second form of error is introduced during reconstruction in a process called *Digital-to-Analogue* conversion and is attributed to the fact that the sum in the interpolation formula is an infinite sum impossible to be implemented in the real-world of circuits. Usually an approximation to the process of interpolation with the sinc function is used.

2.2 Sparse Approximations

Consider the classic sampling theorem and that m equidistant samples of $f(t)$ are gathered in a single vector $\boldsymbol{f} \in \mathbb{R}^m$. Projecting vector $\boldsymbol{f}$ in the Discrete Fourier Transform (DFT) basis would decompose $\boldsymbol{f}$ in a set of coefficients. The magnitude of the said coefficients would then represent the contribution of each of the corresponding sinusoid basis function in signal $\boldsymbol{f}$.

Applying a threshold on the coefficients, i.e., keeping only a subset of the coefficients and setting the rest to a zero value effectively means that the dimensionality of $\boldsymbol{f}$ is forced to a smaller basis. A more sophisticated kind of *thresholding* is met in compression techniques and usually in conjunction with a number of bases that are able to reveal certain aspects of the signal at hand.

2.2.1 Threshold Operators

Consider representing vector $\boldsymbol{f}$ in a basis $\mathcal{B} = \{\boldsymbol{b}_k\}$ with $k = [1, m]$. Approximating $\boldsymbol{f}$ with a subset $\mathcal{G} \subset \mathcal{B}$ of vectors with $s < m$ is referred to as the *s-term approximation*. The quality of the approximation is measured via the s-term approximation error,

$$\epsilon_s = \sum_{i \notin \mathcal{B}-\mathcal{G}} |\langle \boldsymbol{f}, \boldsymbol{b}_i \rangle|^2.$$

Subset $\mathcal{G}$ is usually chosen so as to concentrate most of the energy of $\boldsymbol{f}$. It was proven in [65, Theorem 9.9, p.453] that forming $\mathcal{G}$ via a suitable threshold C will produce the best s-term approximation with respect to ϵ_s for the resulting number of chosen vectors,

$$\mathcal{G} = \{\boldsymbol{b}_k : |\langle \boldsymbol{f}, \boldsymbol{b}_k \rangle| \geq C\}.$$

This threshold operation on the coefficients achieves a signal-specific approxima-

tion scheme which is based on the localised information extracted by transforming a signal to a different domain. This is sometimes referred to as a *non-linear* approximation since subset $\mathcal{G}$ and threshold C depend on the signal.

Sparsity implies the definition of a basis that greatly captures most of the signal's characteristics, e.g., the wavelet coefficients provide good geometrical information on the edges in an image. A *sparsifying* basis means that $s << m$. Subsequently compression, de-noising, filtering and other processing or transmission can be performed efficiently for a sparse signal. This is not always trivial since these adaptive approximations are signal-dependent and not universal for a class of signals. To the engineering community's benefit, most natural signals of interest happen to be sparse in some domain depending on the application.

2.3 Redundant Dictionaries

By slightly changing the terminology let us now move from bases to *dictionaries*. Consider a set of vectors $\{\boldsymbol{h}_k\}_{k\in\mathcal{S}}$ belonging to $\mathbb{R}^m$ (assuming real-valued vectors) with $|\mathcal{S}| > m$. A dictionary $\boldsymbol{H} \in \mathbb{R}^{m\times|\mathcal{S}|}$ is a matrix whose columns are consisted of vectors $\boldsymbol{h}_k$. A discrete signal $\boldsymbol{f}$ can be projected over such a dictionary with its support being $\mathcal{G} = \{\boldsymbol{h}_k : |\langle \boldsymbol{f}, \boldsymbol{h}_k\rangle| > 0\} \subset \mathcal{S}$. At this point it is good to define the ℓ_0 quasi-norm which counts the non-zero components of a vector. Assuming the coefficient vector $\boldsymbol{x}$ for $\boldsymbol{f}$ in $\boldsymbol{H}$, $\|\boldsymbol{x}\|_0 = |\mathcal{G}|$.

In the general case of dictionaries that are *over-complete*, selection of vectors by the application of a threshold on the corresponding coefficients does not yield an optimal approximation with respect to ϵ_s. Result [65, Theorem 12.1,p.612] and [65, Theorem 12.2,p.613] dictate that in the case of over-complete dictionaries minimising the following expression with respect to $\boldsymbol{x}$,

$$\mathcal{L}(\boldsymbol{x}) = \|\boldsymbol{f} - \boldsymbol{f}_s\|^2 + C^2\|\boldsymbol{x}\|_0 \tag{2.1}$$

will return the optimal subset $\mathcal{G}$ and $\boldsymbol{f}_s$ will be the best s-term approximation. This problem collapses to finding the threshold C in case the dictionary is a basis.

It has been proven in [27] that the problem of optimising with respect to the ℓ_0 norm is *NP-hard*. In the computational complexity literature this term describes problems that cannot be solved by existing machines (or algorithms of polynomial complexity) so one *has* to resort to approximations of the problem. To summarise, given a signal $\boldsymbol{f}$ and an over-complete dictionary $\boldsymbol{H}$ one cannot recover the optimal solution $\|\boldsymbol{x}\|_0 = s$ for a given s.

2.3.1 Characterisation of Dictionaries

By introducing more columns than rows, i.e., $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ with usually $m << n$ the sparsity of a representation can be greatly improved. The price to pay is the increased complexity that is required to make the best choice of vectors and also to guarantee *uniqueness* of the representation.

Uncertainty Principle

Studying the uncertainty principle helps in gaining some intuition regarding the representation of a signal in a dictionary. The uncertainty principle states that a particle being described by two complementary variables cannot be described with arbitrary accuracy in both. For example when analysing a signal in time and frequency it cannot be compactly represented in both domains, i.e. a sinusoid is spread in the time domain but it becomes a Dirac function in the Fourier domain.

A very good a example from [36] explains this situation for dictionaries. Assume a dictionary $\boldsymbol{H} = [\boldsymbol{\Psi}, \boldsymbol{\Omega}] \in \mathbb{R}^{m \times n}$ consisted from two orthonormal bases. Also consider a signal $\boldsymbol{b} \in \mathbb{R}^m$. Clearly the following holds:

$$\boldsymbol{b} = \boldsymbol{\Psi}\boldsymbol{y} = \boldsymbol{\Omega}\boldsymbol{x}$$

If $\boldsymbol{\Psi}$ is taken to be the canonical basis and $\boldsymbol{\Omega}$ to be the Fourier basis then $\boldsymbol{y}$ would be the time domain signal while $\boldsymbol{x}$ its Fourier transform. In case $\boldsymbol{\Psi} = \boldsymbol{\Omega}$ then by setting $\boldsymbol{b}$ equal to one of the columns of $\boldsymbol{\Psi}$ would render it sparse in both domains. Deviating from this setting, the sparsity of $\boldsymbol{b}$ in both domains is ruled by the distance between the two bases.

Mutual Coherence

Mutual coherence gives information on how much correlation there is between the columns of a dictionary $\boldsymbol{H} \in \mathbb{R}^{m \times n}$. It is defined as follows:

$$\mu(\boldsymbol{H}) = \max_{1 \leq k \neq l \leq n} \frac{|\boldsymbol{h}_k^T \boldsymbol{h}_l|}{\|\boldsymbol{h}_k\|_2 \|\boldsymbol{h}_l\|_2}$$

and it gives the maximum correlation between two vectors from the same dictionary.

A theorem on the uniqueness of a sparse representation in a dictionary is given below.

Theorem 2 ([36, Theorem 2.5]). *If a system of linear equations $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{y}$ has a solution $\boldsymbol{x}$ obeying $\|\boldsymbol{x}\|_0 < \frac{1}{2}(1 + 1/\mu(\boldsymbol{H}))$, this solution is necessarily the sparsest possible.*

The Spark

The spark of the matrix introduced in [29] is another way to assess the "quality" of a given dictionary. The term stems from the words "sparse" and "rank". *For a dictionary* $\boldsymbol{H}$, spark($\boldsymbol{H}$) *is defined as the smallest number of columns from* $\boldsymbol{H}$ *which are linearly dependent.* As opposed to the rank of a matrix which gives the largest number of columns which are linearly independent. It is easy to verify that the spark is in the range $[2, m+1]$ since the maximum number of independent columns is m and the minimum is 1. This is not a tight upper bound since its exact determination is actually a combinatorial problem.

To be more precise, the spark characterises the dictionary's null-space. The above definition is equivalent to saying that every vector $\boldsymbol{x}$ that belongs in the null-space of $\boldsymbol{H}$ cannot have less than spark($\boldsymbol{H}$) non-zero entries, i.e., $\|\boldsymbol{x}\|_0 \geq \text{spark}(\boldsymbol{H})$. The following is quite similar to Theorem 2.

Theorem 3 ([**36, Theorem 2.4**])**.** *If a system of linear equations* $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{y}$ *has a solution* $\boldsymbol{x}$ *obeying* $\|\boldsymbol{x}\|_0 < \text{spark}(\boldsymbol{H})/2$*, this solution is necessarily the sparsest possible.*

It can further be proven [36, Lemma 2.1] that the mutual coherence can be used to bound the spark. For every dictionary $\boldsymbol{H}$ the following holds:

$$\text{spark}(\boldsymbol{H}) \geq 1 + \frac{1}{\mu(\boldsymbol{H})}.$$

The Restricted Isometry Property

The Restricted Isometry Property (RIP) is another useful metric for dictionaries and was primarily introduced in [17]. Consider the dictionary $\boldsymbol{H} \in \mathbb{R}^{m \times n}$, positive integers $s \leq m$ and constant $0 \leq \delta_s \leq 1$. Also assume all subsets $\mathcal{I} \subset [1, n]$ with $|\mathcal{I}| \leq s$. The dictionary satisfies the s-RIP with δ_s being the smallest constant for which the following holds:

$$(1 - \delta_s)\|\boldsymbol{c}\|_2^2 \leq \|\boldsymbol{H}_{\boldsymbol{\mathcal{I}}}\boldsymbol{c}\|_2^2 \leq (1 + \delta_s)\|\boldsymbol{c}\|_2^2$$

where $\boldsymbol{H}_{\boldsymbol{\mathcal{I}}}$ is the sub-matrix formed by the columns indexed by $\mathcal{I}$ and $\boldsymbol{c} \in \mathbb{R}^s$.

This definition allows us to measure how close to orthonormal are subsets of vectors from a dictionary. Simply put, a matrix formed by any chosen subset of at most s columns $\mathcal{I}$ to form another matrix $\boldsymbol{H}_{\mathcal{I}}$ is almost orthonormal. When $\boldsymbol{H}_{\mathcal{I}}$ acts on a vector $\boldsymbol{c}$ it will alter its magnitude to at most $(1 \pm \delta_s)\|\boldsymbol{c}\|_2^2$, almost like an isometric transformation. Just like in the case of the spark, the RIP constant can

be bounded by the mutual coherence. It can be easily proven that:

$$\delta_s \leq (s-1)\mu(\boldsymbol{H}).$$

It is fair to say that compared to the mutual coherence, the RIP gives a more strict bound on a dictionary since it measures how stable the subsets of the dictionary are rather than the correlation between two vectors. Hence the RIP is a more informative measure. A quite straightforward result from the definition of the RIP is the following,

Lemma 1 ([**17, Lemma 1.3**]). *Assume $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ satisfies the RIP with $\delta_{2s} < 1$ then any signal $\boldsymbol{f} \in \mathbb{R}^m$ with a sparse decomposition $\boldsymbol{x}$ in $\boldsymbol{H}$ satisfying $\|\boldsymbol{x}\|_0 \leq s$, can be exactly represented in $\boldsymbol{H}$.*

Without getting into further technical details one can deduce that matrices obeying the RIP for values of s close to n is an extremely useful attribute.

2.3.2 Incoherent Dictionaries

The theorems mentioned in the previous subsection despite being rudimentary are informative and help with intuition. The existence of a solution is discussed but they do not suggest a way of arriving at one. This is discussed in Section 2.4. The theorems simply suggest that a sparse approximation (as defined in Chapter 2) is possible depending on the coherence of the vectors in the dictionary.

Assume two dictionaries $\boldsymbol{H}$ and $\boldsymbol{\Psi}$ with both of them being orthonormal bases for $\boldsymbol{f} \in \mathbb{R}^m$. In this particular case the mutual coherence of the two bases can be shown to satisfy:

$$\frac{1}{\sqrt{m}} \leq \mu([\boldsymbol{H}, \boldsymbol{\Psi}]) \leq 1.$$

If $\boldsymbol{H}$ is the identity basis and that $\boldsymbol{\Psi}$ is the Fourier basis then it can be verified that the minimum mutual coherence is $\mu([\boldsymbol{H}, \boldsymbol{\Psi}]) = \frac{1}{\sqrt{m}}$.

The minimum mutual coherence of a dictionary constituted of the time and frequency domain bases allows for exact sparse representation of a sparse signal in only one of the domains. It was shown in [30] that random bases with independent, identically distributed entries drawn from the Gaussian distribution of certain variance also have small mutual coherence with *any* fixed basis. This means that a sparse signal in domain $\boldsymbol{H}$ must be spread out in domain $\boldsymbol{\Psi}$ (with which $\boldsymbol{H}$ is incoherent). The fact that $\boldsymbol{H}$ can be generated at random brings us to the next section of this chapter which exploits several results to establish an efficient scheme of sampling sparse signals.

2.4 Compressed Sensing

The incoherence between a sparsity-promoting dictionary and a random sensing domain means that signals can be sparsely represented in one domain while being spread out in the other one. Thus it is possible to acquire information on the sparse components of the signal in the representation domain from random projections in the sensing domain. Based on this fact it is then possible to sample and reconstruct sparse signals with a number of samples far smaller than what is proposed by the Shannon sampling theorem.

Assume that a signal is sampled traditionally at a sufficient rate and then a sparse approximation is obtained in some dictionary by applying the threshold operator. The signal reconstruction provided by Compressed Sensing is at least as good as that sparse approximation. It is thus possible to acquire an adaptive approximation of a given signal *without* adapting the measurement process itself to the signal.

2.4.1 Random Acquisition

Consider an operator described by matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n \times m}$ acting upon the discrete signal $\boldsymbol{f}$,

$$\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{f}$$

giving n linear combinations at its output. In a real-world implementation this would imply an analogue mixing of a natural signal and then subsequent sampling by an analogue-to-digital converter. Usually in Compressed Sensing this operator is *random*, remotely resembling spread-spectrum communications' techniques where a wide-band analogue signal is being modulated by a pseudo-random sequence to spread its content all the way down to the baseband. In Compressed Sensing it is assumed that $n << m$.

It is then assumed that $\boldsymbol{f}$ admits a sparse representation in some domain, $\boldsymbol{f} = \boldsymbol{\Psi x}$ hence the random measurements can be written as:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{\Phi \Psi x} \\ &= \boldsymbol{H x} \end{aligned} \tag{2.2}$$

If the dimensionality of $\boldsymbol{y} \in \mathbb{R}^n$ is sufficiently large then it is possible to recover the signal's sparse support $\boldsymbol{x}$ in domain $\boldsymbol{\Psi}$ (and subsequently $\boldsymbol{f}$). More importantly the acquisition system $\boldsymbol{\Phi}$ is completely agnostic of the signal's particularities in domain $\boldsymbol{\Psi}$.

A word on the term sparse signals

So far the term *sparse* signal has been mentioned with the notion that it can be represented by only a few vectors when decomposed in some suitable domain. Referring to a signal as being s-sparse it is meant that its support $\boldsymbol{x}$ does not have more than s non-zero components. This differs slightly from the definition of the best s-term approximation of a signal which is acquired by keeping only the s largest contributions from an orthonormal dictionary (see Subsection 2.2.1).

The quality of a signal which is sampled compressively depends on whether the signal can be well approximated by a an s-sparse support and more specifically on the decay of the rest of the non-zero components of the support. It was proven in [16] that in case the support has s non-zero components then the support can be recovered exactly.

2.4.2 Acquisition Matrices

Even without an algorithm to perform the sparse support recovery the importance of sensing matrix $\boldsymbol{H} \in \mathbb{R}^{n \times m}$ is recognised. Recovering the correct sparse support for a set of measurements $\boldsymbol{y}$ is greatly affected by the coherence between the sparsity basis $\boldsymbol{\Psi}$ and the sensing system $\boldsymbol{H}$. The way matrix $\boldsymbol{H}$ is generated greatly affects the computational and memory requirements of a reconstruction algorithm since it has to be sufficiently large. Favourable matrices are those which produce highly incoherent dictionaries but also admit efficient implementations.

Gaussian and Bernoulli Random Matrices

The entries of a Gaussian matrix $\boldsymbol{H} \in \mathbb{R}^{n \times m}$ are drawn form the Gaussian distribution, $\mathcal{N}(0, n^{-1})$, hence its columns can be seen as realisations of white noise processes. It has been shown in [18, 28] that $\boldsymbol{H}$ satisfies the RIP with exponentially increasing probability as n increases, for Gaussian $\boldsymbol{H}$ and any orthonormal basis $\boldsymbol{\Psi}$, when the following holds:

$$n \geq C \cdot s \cdot log\left(\frac{m}{s}\right)$$

where C is a constant. This expression relates the number of measurements n, the dimensionality of the sparsity basis m and the number of non-zeroes s in the sparse support. By constructing a random Gaussian matrix of appropriate dimensions the RIP holds with a certain probability. Given a sufficiently large n, $\boldsymbol{H}$ satisfies the RIP almost for certain.

The same claim can be made for when the entries of matrix $\boldsymbol{H}$ are drawn from the Bernoulli distribution, i.e. they take values $\pm\frac{1}{\sqrt{n}}$ with probability of $\frac{1}{2}$. In the case of the Bernoulli distribution the constant C is larger hence more measurements

are needed but operations become computationally cheaper since multiplications become additions. In both cases memory requirements are $\mathcal{O}(n \cdot m)$.

Sub-sampled Projectors

In this class of sensing matrices dictionaries are constructed by assuming structured operators such as the Discrete Fourier Transform matrix. This is achieved by choosing vectors uniformly at random (without replacement). It was proven in [18] that the RIP for a sub-sampled Fourier matrix $\boldsymbol{H}$ holds with exponentially increasing probability as m increases when the following holds:

$$n \geq C \cdot s \cdot \left(log\left(\frac{m}{s}\right)\right)^4$$

where C is a constant. Of course in this case constant C becomes dependent on the mutual coherence between the two orthonormal bases.

Compared to the Gaussian and Bernoulli matrices, these random sensing matrices are far more efficient to implement and to manipulate due to their structured nature. Usually fast algorithms exploit this structure to accelerate computations, i.e. for Fourier or wavelet bases. The price to pay of course is a larger lower bound on the number of measurements because of the dependence on the mutual coherence between $\boldsymbol{H}$ and $\boldsymbol{\Psi}$ something which Gaussian/Bernoulli matrices do not suffer from.

2.5 Reconstruction Algorithms

So far no mention has been made on any method to recover a sparse approximation from the samples. The problem is written as follows

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \text{ so that } \boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{2.3}$$

which is computationally impossible to solve in polynomial time [27]. It is assumed that $\|\boldsymbol{x}\|_0 = s$.

The approximations to such a solution can usually be divided into two categories; greedy algorithms (or pursuits as usually mentioned in the bibliography) and convex relaxation methods. For both categories solutions can be found which coincide with the global minimum of Equation (2.3). This is translated into constraints regarding the value of s and the specific dimensionality of the problem at hand. Success cannot be guaranteed for every possible case since that would bring us back to the problem of Equation (2.3).

2.5.1 Convex Relaxation

The main difficulty with the optimisation problem described in (2.3) is the non-convexity of the ℓ_0 term. Convex problems make optimisation easier and are quite attractive in the signal processing area. Consider using the ℓ_2 norm in (2.3), then the optimisation problem becomes strictly convex and admits a closed form solution.

Replacing with the ℓ_1 norm gives rise to what is referred to in the bibliography as *Basis Pursuit* (BP) and can be seen more like a principle rather than an algorithm. It is stated as

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ so that } \boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{2.4}$$

and even though it is *not* strictly convex it has some appealing properties. It is easy to visualise two optimal solutions with the same ℓ_1 norm something which means that there exist infinite solutions to this problem. Despite the fact that uniqueness cannot be guaranteed for non-strictly convex problems the two following attributes can be guaranteed: first the solutions belong in a bounded convex set and second there exists at least one solution with a maximum of n non-zero components. An intuitive explanation can be found in [36]. There is a natural preference of the ℓ_1 norm towards sparse solutions.

This modification of Equation (2.3) results in a well-posed problem which can be solved by *Linear Programming* [17]. Then the problem of sparse reconstruction can be solved by interior point algorithms, the simplex method and others. The major drawback of such algorithms is their sophisticated nature and their sometimes prohibitively complex implementation. The FOCUSS algorithm proposed in [45] attempts an approximation to Basis Pursuit which admits a simple implementation and avoids local minima but it does not guarantee convergence to a minimum and has the danger of dwelling to a fixed point.

Performance Guarantees

In [16, 15] it was proven that under certain conditions Basis Pursuit and ℓ_0 minimisation arrive at the same solution. This does not suggest that an algorithm to solve Basis Pursuit behaves in the same way like an algorithm to solve ℓ_0.

Theorem 4 ([16]). *Consider the solution $\boldsymbol{x}^*$ given by Basis Pursuit. Further assume that matrix $\boldsymbol{H}$ satisfies the RIP with $\delta_{2s} < \sqrt{2} - 1$ then it holds that*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leq \frac{C}{\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{x}_s\|_1 \qquad \|\boldsymbol{x}^* - \boldsymbol{x}\|_1 \leq C \|\boldsymbol{x} - \boldsymbol{x}_s\|_1$$

where C is a constant and $\boldsymbol{x}$ is the actual solution. Vector $\boldsymbol{x}_s$ is formed by zeroing all but the s largest entries of $\boldsymbol{x}$.

The bound has been further improved to $\delta_{2s} < 0.4652$ in [41]. The theorem suggests that if a signal has an exactly s-sparse support then it is recovered exactly given the sufficient condition on the RIP of the sensing matrix. In any other case it returns the best s-term approximation. Also unlike previous results [13] this theorem is deterministic since is suggests exact recovery without probability of error.

2.5.2 Greedy Algorithms

The brute force strategy silently proposed by Equation (2.3) gives way to a constructive way of finding a solution. Algorithms of this category start off with an empty support set and then based on a selection criterion, column indices are added or removed. These algorithms focus on the fact that the problem of recovering the solution via Equation (2.3) can be solved by first identifying the correct support and then recovering the magnitudes by least squares.

The Orthogonal Matching Pursuit

A widely used algorithm is the Orthogonal Matching Pursuit (OMP) which made its appearance in the engineering field well before the Compressed Sensing. The OMP selects the elements of the support sequentially based on the correlation of the residual signal with the remaining vectors of the dictionary. It starts off with the residual being equal to the measurements vector $\boldsymbol{y}$. The contribution of the previously selected support vectors is eliminated from the residual and the new residual is formed. The steps for the OMP are given below as it will partake in the following discussions.

Algorithm 1 Orthogonal Matching Pursuit

Input: s, $\boldsymbol{H}$, $\boldsymbol{y}$

Initialise:

1. $\mathcal{T}^0 = \emptyset$
2. $\boldsymbol{y}_r^0 = \boldsymbol{y} - \boldsymbol{H}_{\mathcal{T}^0}\boldsymbol{H}_{\mathcal{T}^0}^{\dagger}\boldsymbol{y}$.

Iteration l:

1. $\mathcal{T}^l = \mathcal{T}^{l-1} \cup \{$ index corresponding to the largest magnitude entry in vector $\boldsymbol{H}\boldsymbol{y}_r^{l-1}\}$.
2. Calculate residual: $\boldsymbol{y}_r^l = \boldsymbol{y} - \boldsymbol{H}_{\mathcal{T}^l}\boldsymbol{H}_{\mathcal{T}^l}^{\dagger}\boldsymbol{y}$.
3. If $l = s$ quit.

Output:

1. The estimated signal $\hat{\boldsymbol{x}}$ satisfying $\hat{\boldsymbol{x}}_{\{1,\cdots,m\}-\mathcal{T}^l} = \boldsymbol{0}$ and $\hat{\boldsymbol{x}}_{\mathcal{T}^l} = \boldsymbol{H}_{\mathcal{T}^l}^{\dagger}\boldsymbol{y}$.

In [91] it was proven that given certain requirements on the mutual coherence of $\boldsymbol{H}$ the OMP recovers all s-sparse signals exactly.

Theorem 5 ([91]). *For $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ with $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ and $(n < m)$, if the following holds*

$$\|\boldsymbol{x}\|_0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{H})}\right),$$

then the OMP algorithm recovers all s-sparse signals exactly.

After some basic calculations it is evident that the OMP places more strict bounds on the RIP than Basis Pursuit. Actually BP shows superior performance empirically but the OMP enjoys extremely simple implementations and far lower computational requirements. The computational complexity of this type of pursuit algorithm is ruled by the number of iterations needed for exact reconstruction. For the OMP this complexity is roughly $\mathcal{O}(s \cdot n \cdot m)$ since it recovers an s-sparse signal in exactly s iterations.

Other Greedy Pursuits

The OMP has been extended to the Regularised OMP (ROMP) [73] and the Stagewise OMP (StOMP) [34]. The algorithms follow a similar path with the OMP by selecting several candidate columns from $\boldsymbol{H}$ for inclusion in the support set based on their correlation values with the residual. Those vectors that are deemed reliable based on a given criterion are eventually added to the support set. The pursuit continues until a finishing criterion is met. Computational complexity is also lower than Basis Pursuit. The ROMP algorithm reconstructs all s-sparse signals exactly given that $\boldsymbol{H}$ satisfies the RIP with $\delta_{2s} \leq \frac{0.06}{\sqrt{s}}$. The RIP requirements remain more strict than those for Basis Pursuit given in Theorem 4 and are also dependent on s.

Another type of greedy algorithms promote sparsity in the solutions by applying hard thresholds like the one in [11]. These algorithms admit extremely low computational effort and show performance guarantees comparable to other more sophisticated sparse reconstruction algorithms mentioned below. Indicatively the algorithm in [11] recovers all s-sparse signals exactly given that the RIP holds with $\delta_{3s} < \frac{1}{\sqrt{32}}$. Further discussion on this type of algorithms does not take place here since the present text does not negotiate any of their properties.

Subspace Pursuit

The Subspace Pursuit (SP) introduced in [26] attempts to minimise the gap between the superior performance of Basis Pursuit and the low computational complexity of greedy pursuits. The Compressive Sampling Matching Pursuit (CoSaMP) algorithm introduced in [72] admits similar analysis and performance guarantees as the SP. The key point behind these algorithms is that once elements are included in the support set they are then allowed to be excluded if deemed appropriate. This

is possible because of an added backtracking step. Collocated in Algorithm 2 are the steps for the SP algorithm since this specific algorithm will take part in the discussions in the chapters to follow.

The authors in [26] provide the following theorem regarding the performance of SP.

Theorem 6 ([26]). *For $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ with $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ and $(n < m)$, if the following holds*

$$\delta_{3k} < 0.205,$$

then the SP algorithm recovers all s-sparse signals exactly in a finite number of iterations.

Algorithm 2 Subspace Pursuit

Input: s, $\boldsymbol{H}$, $\boldsymbol{y}$

Initialise:

1. $\mathcal{T}^0 = \{s$ indices corresponding to the largest magnitude entries in the vector $\boldsymbol{H}\boldsymbol{y}\}$.
2. $\boldsymbol{y}_r^0 = \boldsymbol{y} - \boldsymbol{H}_{\mathcal{T}^0}\boldsymbol{H}_{\mathcal{T}^0}^{\dagger}\boldsymbol{y}$.

Iteration l:

1. $\tilde{\mathcal{T}}^l = \mathcal{T}^{l-1} \cup \{s$ indices corresponding to the largest magnitude entries in the vector $\boldsymbol{H}\boldsymbol{y}_r^{l-1}\}$.
2. Set $\boldsymbol{x}_p = \boldsymbol{H}_{\tilde{\mathcal{T}}^l}$.
3. $\mathcal{T}^l = \{s$ indices corresponding to the largest elementsof $\boldsymbol{x}_p\}$.
4. $\boldsymbol{y}_r^l = \boldsymbol{y} - \boldsymbol{H}_{\mathcal{T}^l}\boldsymbol{H}_{\mathcal{T}^l}^{\dagger}\boldsymbol{y}$.
5. If $\|\boldsymbol{y}_r^l\| < \|\boldsymbol{y}_r^{l-1}\|$, let $\mathcal{T}^l = \mathcal{T}^{l-1}$ and quit.

Output:

1. The estimated signal $\hat{\boldsymbol{x}}$ satisfying $\hat{\boldsymbol{x}}_{\{1,\cdots,m\}-\mathcal{T}^l} = \boldsymbol{0}$ and $\hat{\boldsymbol{x}}_{\mathcal{T}^l} = \boldsymbol{H}_{\mathcal{T}^l}^{\dagger}\boldsymbol{y}$.

2.5.3 Measurement Perturbations

So far it has been assumed that the compressed measurements did not suffer from the effects of noise. Here it is assumed that the measurements are corrupted with random noise usually white. The model described in Equation (2.2) becomes

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{2.5}$$

where the entries of vector $\boldsymbol{n}$ are independent and identically distributed drawn from $\mathcal{N}(0, \sigma^2)$. The results that have been presented only consider the noiseless case. The authors of the respective work have provided the noisy counterparts of

their theorems with the corresponding error bounds and performance guarantees. In this text further results are not presented since a complete study on Compressed Sensing is not the goal of the current text.

Basis Pursuit De-noising

The optimisation problem posed by the Basis Pursuit in Equation (2.4) is transformed into the following:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ so that } \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2 \leq \epsilon \tag{2.6}$$

which is known as Basis Pursuit Denoising (BPDN). The value of parameter ϵ depends on the noise variance which is assumed to be bounded. Actually when the same problem is transformed into a Lagrange unconstrained optimisation problem it becomes what is known to the machine learning community as the Least Absolute Shrinkage and Selection Operator (LASSO). The LASSO problem statement is the following

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + C\|\boldsymbol{x}\|_1. \tag{2.7}$$

Constant C is proportional to the noise variance. It is easy to verify that when the noise variance (or C) goes to zero then the problem goes back to solving Basis Pursuit.

The notions of the spark, the mutual coherence and the RIP are still extensively used in order to assess the stability of the solutions for each algorithm with the RIP being the most flexible tool for this task. Indicatively one result for BPDN from [16] is mentioned in the following theorem. The analyses for greedy algorithms like the OMP, SP, CoSaMP and other all demonstrate results of similar nature.

Theorem 7 ([16])**.** *Consider the solution $\boldsymbol{x}^*$ given by BPDN. Given that $\boldsymbol{H}$ satisfies the RIP for $\delta_{2s} < \sqrt{2} - 1$ then the following holds*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leq \frac{C_1}{\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{x}_s\|_2 + C_2\epsilon$$

where C_1, C_2 are constants and $\boldsymbol{x}$ is the actual solution. Vector $\boldsymbol{x}_s$ is formed by zeroing all but the s largest entries of $\boldsymbol{x}$.

This extends the noiseless case stated in Theorem 4 and sates that BP solves the sparse reconstruction problem in the noisy case with an error bounded by the amount of noise and the error if the support had exactly s non-zero components. Thus noise is handled in a controlled manner.

The Dantzig Selector algorithm

The Dantzig Selector (DS) algorithm is mentioned here for completeness. It was introduced in [14] and it serves as an alternative to BPDN. The algorithm attempts to solve the following problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ so that } \|\boldsymbol{H}^T(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})\|_\infty \leq \epsilon \tag{2.8}$$

where again ϵ depends on the noise variance.

The first thing to notice when compared to BPDN is that the DS has an additional requirement for the residual signal. In a way it does not only constrain the residual to be within the noise level but also allows for the residual to be structured, i.e. correlated with the columns of $\boldsymbol{H}$. The DS can also be formulated as a Linear Program and be solved by one of the available methods.

2.6 Conclusion

Representing a signal in the digital world accurately is only part of the story. An analogue signal undergoes discretisation and can be exactly reconstructed as long as a restriction on its frequency content is imposed. Taking this a little bit further, the ultimate goal of the sampling process is to unravel that part of the signal which is particularly important. The classic sampling theorem does not take into account the inherent sparsity of the signal, i.e., the fact that not all transform coefficients have equal weight. From this standpoint it is clear that in order to recover the signal, the number of samples required should not be independent of the number of important frequencies (if the Fourier domain is assumed).

The modern sampling theories that all fall under the title of Compressed Sensing, suggest a different sampling procedure altogether; instead of sampling at rates comparable to the associated bandwidths to only take as many samples required for the actual sparse components to be recovered. This comes with a number of supporting theorems that set the foundations. Theoretical results that prove such a feat is possible and under which circumstances. Much like the traditional sampling theorem by Shannon. Moreover, due to the particularities, recovery comes at the cost of a sophisticated algorithm as opposed to the interpolation formula. Again, researchers have been able to provide a vast set of algorithms along with their intricate performance guarantees.

Chapter 3

Hardware Architectures for Compressed Sensing

There seems to be a discrimination in the bibliography regarding the actual random sampling of an analogue signal and then its subsequent recovery. This etiquette is adopted here and this chapter is divided into two sections; one for random sampling and one for sub-Nyquist sampling architectures.

This chapter serves as a short summary of some of the most prominent machinery encountered in the area of random sampling and analogue sparse signal recovery. Some of the most commonly met random sampling schemes in the bibliography are presented with minor technical details. These are the Multi-coset sampling scheme, the Co-prime sampling scheme and the Discrete Random Sampling framework.

Since these sampling schemes are not necessarily tied to a reconstruction procedure but to rather generic sparse recovery algorithms, three of the most popular analogue sparse signal sampling and recovery architectures are presented that have been proposed for actual hardware implementations. These include the Random Filters approach, the Random Demodulator and the Modulated Wideband Converter.

3.1 Non-uniform Sampling Schemes

In previous chapters it was shown that the Compressed Sensing framework requires for the sampling operator to introduce a certain amount of *randomness* into the acquired samples. This process is related to what has been discussed in Subsection 2.4.1. Let us put theory to the side for a while and see how this *randomness* can actually be introduced in some real-world scenarios.

Most of the time it is assumed that the sampling domain is that of time even though these techniques apply to the space domain as well. Randomness is usually introduced by sampling a signal at non-uniform sampling intervals.

3.1.1 Multi-coset Sampling

Multi-coset sampling was studied in [38]. The authors propose the use of multiple uniform sampling branches at rates well below the Nyquist rate but with different phase offsets and time delays. The scheme is depicted in Figure 3.1.

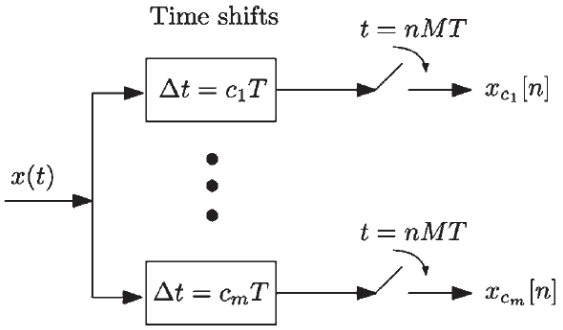


Figure 3.1: *Multi-coset non-uniform sampling scheme.*

In the diagram the delays are chosen so that $0 \leq c_i \leq M - 1$ where M is a positive integer. A choice of $m < M$ distinct such integer delays is made. Then the samples for each sampling branch are naturally written as,

$$x_{x_i}[n] = x(nMT + c_i T)$$

where $1/T$ is the Nyquist rate. Based on the above the average sampling rate is $\frac{m}{MT}$ which is lower than the Nyquist rate.

3.1.2 Co-prime Sampling

Co-prime sampling was introduced in [95] and is briefly summarised in Figure 3.2.

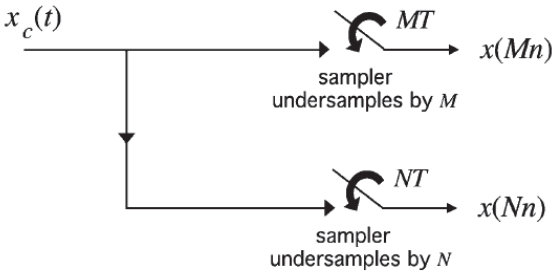


Figure 3.2: *Co-prime sampling scheme.*

The diagram depicts the original formulation where two sampling branches are employed. The scheme can extended to more than two branches. The key idea is that the under-sampling factors M, N are co-prime integers. The authors base their

results in some fundamental properties of co-prime numbers to establish a sampling framework where a signal is sampled sparsely but several aspects of the signal like the spectrum is calculated at a significantly higher resolution. The average sampling rate is $\frac{1}{MT} + \frac{1}{NT}$.

3.1.3 Discrete Random Sampling

The authors of [63] leverage the work of previous authors to establish some very useful results for sampling an analogue signal at randomly selected points in time which reside on a predefined uniform grid. The sampling points are taken in the fashion, of the diagram in Figure 3.3.

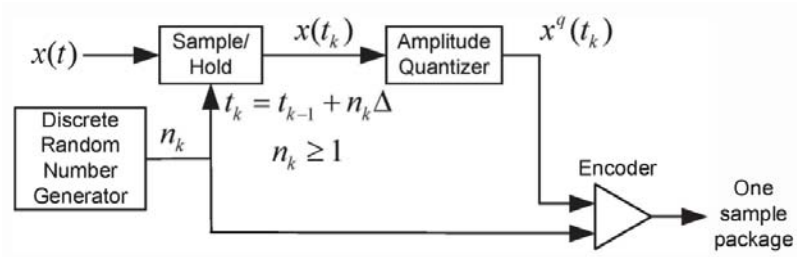


Figure 3.3: *Additive Random sampling scheme.*

In the functional diagram Δ is the uniform spacing of the predefined grid and n_k is a discrete random variable chosen based on some chosen distribution.

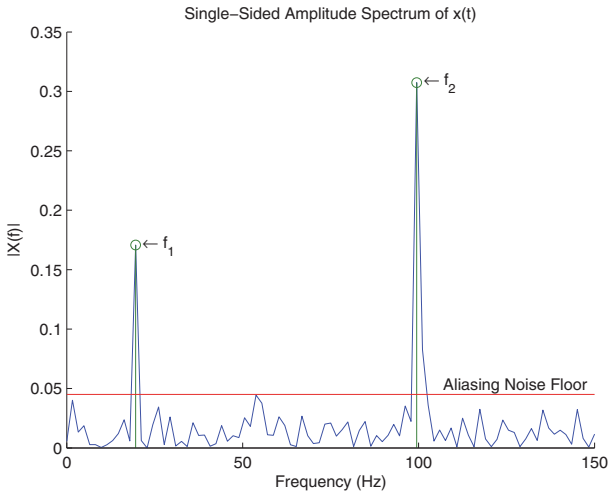


Figure 3.4: *Example of the ARS scheme.*

Sampling in such a way (and on non-uniform time intervals in general) results in that the sampled spectrum to be consisted of an aliasing noise-floor in addition to the spectral content of the sampled signal as shown in Figure 3.4. The theoretical results include the exact relationship between the aliasing noise floor and the desired frequency content with respect to the distribution function of the sampling points.

3.2 Sub-Nyquist Sampling Architectures

Here three well known architectures are summarised. These architectures were introduced as complete sampling and reconstruction systems rather than just random sampling schemes like the ones presented in the previous section. The Random Filters approach can be taken as a theoretical approach since it only considers discrete time signals as input but helps build intuition. The Random Demodulator and the Multi-band Wideband Converter are two architectures aiming at sub-Nyquist sampling of analogue signals with the latter being the first architecture proposed to be backed up by a hardware prototype.

3.2.1 Random Filters

The Random Filters technique was introduced in [93] as a practical means to sample and reconstruct a signal at sub-Nyquist rates. As the authors state in their paper, this method was analysed and empirically tested solely for *discrete* signals. Even though this method is not applicable in real-world it is a very good starting point for understanding how sub-Nyquist sampling and sparse recovery algorithms are tied in together and how the reconstruction process is affected by the introduction of randomness.

The method assumes that a discrete signal $\boldsymbol{s}$ of length d can be compactly represented in a basis $\boldsymbol{\Psi}$ with only $m << d$ components, i.e, signal $\boldsymbol{s}$ is $m - sparse$. In mathematical language:

$$\boldsymbol{s} = \boldsymbol{\Psi\theta}$$

where vector $\boldsymbol{\theta}$ has m non-zero entries representing the coefficients of $\boldsymbol{s}$ in basis $\boldsymbol{\Psi}$.

Compressed Sensing theory dictates that by acquiring a sufficient number N of linear measurements of $\boldsymbol{s}$ in a randomised manner can yield exact recovery of any $m - sparse$ signal,

$$\boldsymbol{y} = \boldsymbol{\Phi s}$$

where $\boldsymbol{y}$ is the N-dimensional measurement vector and $\boldsymbol{\Phi}$ is a randomly sourced measurement matrix. The relationship between the number of measurements required for exact recovery and the type of matrix employed has been extensively studied in the previous chapter.

Recovery aims to find an approximation to the well known problem,

$$\hat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 \text{ so that } \boldsymbol{y} = \boldsymbol{\Phi\Psi\theta}.$$

The recovery process has to be taken care of by a *non-linear* algorithm such as the Orthogonal Matching Pursuit (OMP), the Subspace Pursuit [26] or CoSaMp

[72]. Exact recovery can also be accomplished with other computationally expensive algorithms based on Linear Programming.

Filters with Random Tap Weights

The authors assume a filter $\boldsymbol{h}$ of length B whose entries are randomly sourced independently from the Gaussian distribution $\mathcal{N}(0,1)$. This choice is not restrictive but facilitates understanding without compromising generality. The sampling process is then described likewise:

$$\boldsymbol{y} = \boldsymbol{\Phi s}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} h_4 & h_3 & h_2 & h_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & h_4 & h_3 & h_2 & h_1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & h_4 & h_3 & h_2 & h_1 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{15} \end{bmatrix} \tag{3.1}$$

where it is assumed for illustration purposes that $d = 16, N = 5$ and $B = 4$. Matrix $\boldsymbol{\Phi}$ is built so that each of its rows is equal to the previous one shifted by $\lfloor d/N \rfloor$. This is described compactly in the following diagram:

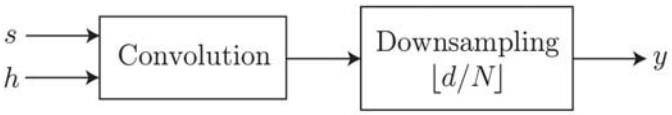


Figure 3.5: *The Random Filter Architecture*

The process described by Equation (3.1) can be seen as convolution of the signal with the filter followed by a down-sampling operation.

OMP-based Recovery Algorithm

The recovery procedure which aims at finding a suitable sparse vector $\hat{\boldsymbol{\theta}}$ is carried out by the algorithm in Algorithm 3.

Algorithm 3 Random Filters Recovery Algorithm

Input: $\boldsymbol{\Phi\Psi}$, $\boldsymbol{y}$
Initialise: $\boldsymbol{r}_0 = \boldsymbol{y}$
Iteration $l = 1..N$:

1. Find the column i_l of $\boldsymbol{\Phi\Psi}$ such that

$$i_l = \arg\max_i |\langle \boldsymbol{r}_{l-1}, (\boldsymbol{\Phi\Psi})_i \rangle|.$$

2. Compute the new residual

$$\boldsymbol{r}_l = \boldsymbol{y} - \boldsymbol{P}_l \boldsymbol{y}$$

where $\boldsymbol{P}_l$ is the orthogonal projector onto the span of the l columns chosen from $\boldsymbol{\Phi\Psi}$.

Output:

1. Columns $\{i_l\}$ and coefficients $\{\hat{\theta}_{i_l}\}$ such that

$$\boldsymbol{P}_N \boldsymbol{y} = \sum_{l=1}^{N} \hat{\theta}_{i_l} \left(\boldsymbol{\Phi\Psi}\right)_{i_l}.$$

The algorithm is based on the OMP algorithm but accepts several enhancements as the authors mention because of the special structure of matrix $\boldsymbol{\Phi}$. These alternations aim towards reducing the computational complexity. The authors test the recovery algorithm on several cases of sparse signals including toy problems based on signals sparse in the time domain, the Fourier domain and the Haar wavelet domain. The empirical findings will not be discussed in this text and the interested reader is redirected to the relevant bibliography for a more elaborate description.

3.2.2 The Random Demodulator

The authors in [92] present a sub-Nyquist signal acquisition system namely the Random Demodulator. This is a result of previous work of other researchers which will not be presented here. This authors present a long study of the system covering the separate components of the system, practical aspects and the signal model that is followed. The work is supported by a rigorous set of empirical evidence and a detailed theoretical analysis on the performance guarantees of the system. These include the minimum sampling rate required for exact recovery versus the sparsity of the signal.

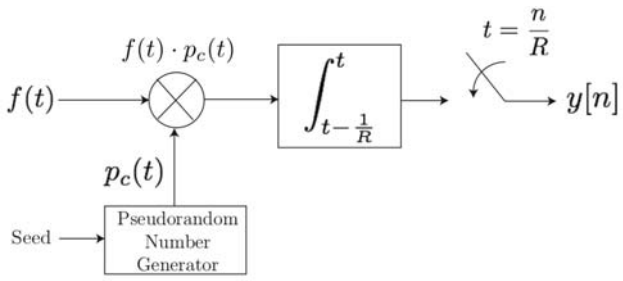


Figure 3.6: *The Random Demodulator Architecture*

The architecture is shown in Figure 3.6. The band-limited analogue signal $f(t)$ is *demodulated* by a pseudo-random sequence of ± 1, called the *chipping sequence*. The rate at which this sequence alternates is *at or above* the Nyquist rate of the signal W Hz. The output of the mixer is then low-pass filtered and down-sampled by a factor R. In reality the low-pass filter is an accumulator which sums the mixed signal for $1/R$ seconds. The output is then digitised by an analogue-to digital converter. The key idea is that the rate R at which the signal is sampled is far lower than the Nyquist rate W. The value of R depends on the number of active frequencies in the analogue signal $f(t)$.

The process described above introduces randomness in the sampling process in a different way than Random Filters since it is aimed in actual sampling of analogue signals. The authors note that the hardware needed is not specialised in any way and is readily available.

Signal Model

The assumed signal model is mathematically described as,

$$f(t) = \sum_{\omega \in \Omega} a_\omega e^{-2\pi\omega t} \quad for\ t \in [0, 1)$$

where $\Omega \in \{0, \pm 1, \pm 2, \cdots, W/2\}$ is the set of the $K << W$ active frequencies and a_ω are the corresponding amplitudes. The time interval has been normalised for ease of exposition.

In real-world applications the above signal model is not always met since it implies that the signal only contains harmonics residing exactly on the Fourier grid. Unfortunately this is not always the case. The authors propose a classical remedy to this problem, that is to introduce a windowing operation prior to acquiring f.

Operation of the Random Demodulator

It can be proven that the continuous time domain signal $f(t)$ comprising of the frequencies in Ω can be written down as a discrete time signal x_n with the same

frequency content:

$$x_n = \sum_{\omega \in \Omega} s_\omega e^{-2\pi i n\omega/W} \quad n = 0, 1, \cdots, W-1$$

where

$$s_\omega = a_\omega \frac{e^{-2\pi i n\omega/W - 1}}{2\pi i\omega}.$$

The above can then be easily written in matrix form:

$$\boldsymbol{x} = \boldsymbol{F}\boldsymbol{s}$$

where matrix $\boldsymbol{F}$ is the $W \times W$ DFT matrix. The actions of mixing with the chipping sequence and the accumulator can be written also in matrix form:

$$\boldsymbol{D} = \begin{bmatrix} \epsilon_0 & & & \\ & \epsilon_1 & & \\ & & \ddots & \\ & & & \epsilon_{W-1} \end{bmatrix} \quad \boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & & & & & \\ & & & 1 & 1 & 1 & & \\ & & & & & & \ddots & \\ & & & & & & 1 & 1 & 1 \end{bmatrix}$$

where $\{\epsilon_0, \cdots, \epsilon_{W-1}\}$ is the chipping sequence. Matrix $H \in \mathbb{R}^{R \times W}$ resembles the matrix that was encountered in the Random Filters. In the example above it is assumed that $\frac{W}{R} = 3$. The entries of each row are adjusted accordingly for when $\frac{W}{R}$ is not an integer.

The whole process can be written as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{D}\boldsymbol{F}\boldsymbol{s} = \boldsymbol{\Phi}\boldsymbol{s}$$

and matrix $\boldsymbol{\Phi}$ is called the random demodulator matrix. Like with Random Filters, a recovery algorithm needs to be employed to solve

$$\hat{\boldsymbol{s}} = \min \|\boldsymbol{v}\|_1 \text{ so that } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{v}.$$

and find the most suitable sparse vector $\hat{\boldsymbol{s}}$.

3.2.3 The Modulated Wideband Converter

The Modulated Wideband Converter (MWC) was introduced in [68] for wideband sub-Nyquist sampling of analogue signals. The authors aimed at efficient hardware implementation and low computational load achieved through simple recovery algorithms. The architecture is based on a sub-band acquisition system that more or less resembles classical approaches. It is depicted in Figure 3.7. The con-

sidered signal model assumes that $x(t)$ occupies only a small number N of bands in the entire wideband spectrum. B is assumed to be the maximum bandwidth of each band in the frequency content of $x(t)$.

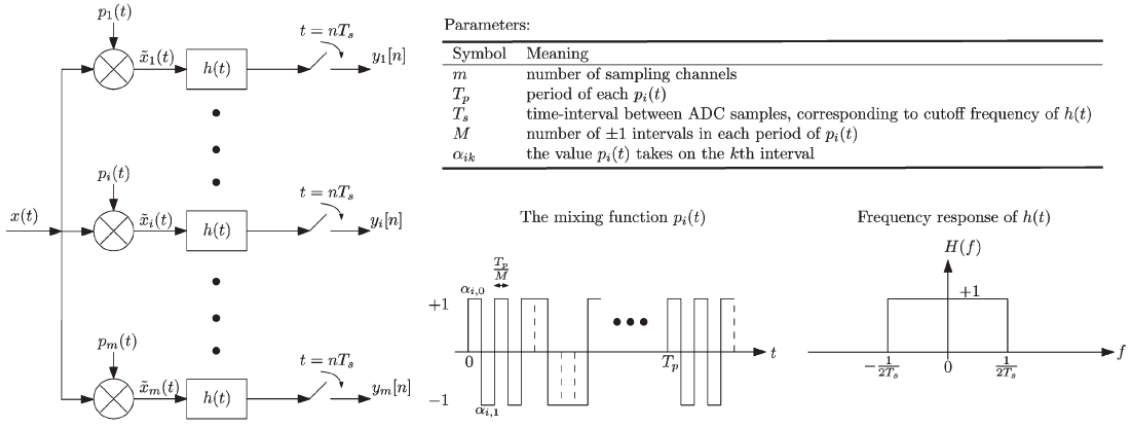


Figure 3.7: *The Modulated Wideband Converter Architecture*

The MWC is consisted of m channels. At each channel the analogue input signal $x(t)$ is mixed with a T_p periodic mixing sequence which in this particular example is a square wave with M as shown in the diagram. The mixing sequence signs are be chosen uniformly at random so that the sequences for each channel are adequately different from each other. The value of $f_p = 1/T_p \geq B$ is chosen so that parts of the spectrum from each band will be aliased down to baseband. The mixed signal is then fed into a low-pass filter with a cut-off frequency of $1/2T_s$. Ideally the frequency response is rectangular. The filtered signal is then sampled by an ADC at a rate of $f_s = 1/T_s$.

The whole process can be written in matrix form as shown below in Figure 3.8

$$\underbrace{\begin{pmatrix} Y_1(e^{j2\pi f T_s}) \\ Y_2(e^{j2\pi f T_s}) \\ \vdots \\ Y_m(e^{j2\pi f T_s}) \end{pmatrix}}_{\mathbf{y}(f)} = \underbrace{\begin{bmatrix} \alpha_{1,0} & \cdots & \alpha_{1,M-1} \\ \vdots & \ddots & \vdots \\ \alpha_{m,0} & \cdots & \alpha_{m,M-1} \end{bmatrix}}_{\mathbf{S}} \underbrace{\begin{bmatrix} | & \cdots & | & \cdots & | \\ \bar{\mathbf{F}}_{L_0} & \cdots & \bar{\mathbf{F}}_0 & \cdots & \bar{\mathbf{F}}_{-L_0} \\ | & \cdots & | & \cdots & | \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} d_{L_0} & & \\ & \ddots & \\ & & d_{-L_0} \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} X(f - L_0 f_p) \\ \vdots \\ X(f) \\ \vdots \\ X(f + L_0 f_p) \end{bmatrix}}_{\mathbf{z}(f)}$$

Figure 3.8: *The MWC operation in matrix form*

Where $Y_i(e^{i2\pi f T_s})$ for $f \in \mathcal{F}_s = \{-f_s/2, +f_s/2\}$ is the Fourier transform of sequence $y_i[n]$, the elements of the $m \times M$ matrix $\boldsymbol{S}$ are the elements of the mixing sequences taken from $\{\pm 1\}$ and matrix $\boldsymbol{F}$ is an $M \times L$ sub-matrix of the $M \times M$ DFT matrix. The constant $L \times L$ diagonal matrix D is defined in the original paper. Further details are omitted from this functional description of the converter. Vector $\boldsymbol{z}(f)$

contains L slices of the spectrum where,

$$L_0 = \left\lceil \frac{f_{Nyq} + f_s}{2f_p} \right\rceil - 1$$
$$L = 2L_0 + 1$$

and represents the unknown spectral content of $x(t)$. The Nyquist rate for $x(t)$ is f_{Nyq}. One of the conditions for recovery presented in the paper is $M \geq L$. More precisely,

$$z_i(f) = X(f + (i - L_0 - 1)f_p), \quad 1 \leq i \leq L, \ f \in \mathcal{F}_s$$

where $X(f)$ is the Fourier transform of $x(t)$.

The theoretical results establish the rules for choosing the parameters for the problem specified above so that the system will function properly and for recovery to be accurate. The choice of $f_p \geq B$ results in that each of the N bands contributes only one non-zero element in $\boldsymbol{z}$. This means that $\boldsymbol{z}(f)$ is $N - sparse$.

Signal Reconstruction

Recovery is based upon recovering the sparsest $\boldsymbol{z}(f)$ for each $f \in \mathcal{F}_s$. Unlike the Random Filters and the Random Demodulator the MWC employs a different approach for sparse recovery in which the support set (the index set of the non-zero entries) and the magnitudes are calculated separately. This is done so as to achieve gains in performance due to the structure of the MWC.

Let us write the process in Figure 3.8 a bit more compactly

$$\boldsymbol{y}(f) = \boldsymbol{A}\boldsymbol{z}(f) \ \ f \in \mathcal{F}_s$$

where $\boldsymbol{A} = \boldsymbol{SFD}$. The algorithm to solve the problem stated above is based on the Orthogonal Matching Pursuit and is called Simultaneous OMP [67]. The algorithm recovers the support set $S = \text{supp}(\boldsymbol{z}(\mathcal{F}_p))$ where $\mathcal{F}_p = \{-f_p/2, +f_p/2\}$. Set S is the union of supports of $\boldsymbol{z}(f)$ for $f \in \mathcal{F}_p$. The way to achieve this is shortly described in the original paper and is based on previous work of the same authors. After S has been recovered one performs,

$$\boldsymbol{z}_S[n] = \boldsymbol{A}_S^{\dagger}\boldsymbol{y}[n]$$
$$z_i[n] = 0, \ \ i \notin S$$

to compute the inverse DTFT of $\boldsymbol{z}(f)$.

3.3 Conclusion

A very quick tour through that small place in hardware-land was taken in this small chapter, that deals with actual ways to implement compressed sensing in the real world. The amount of work that has been put into developing an actual hardware platform is disanalogous to the theoretical work and not many implementations have been reported so far, at the time of writing of course year 2015. The reasons for this can be many such as the daunting *randomness* feature these samplers have to exhibit. Many of the questions a design engineer would have to answer is, "How random does it have to be ?". The answer to this question directly affects the performance of a compressed sampler. Another difficult task is to find ways to implement randomness in the analogue domain, something which is greatly limited by the nature of the signal to be sampled. Researchers and engineers have worked together to find ways to understand how to overcome these issues and make the most benefit out of sparsity.

Chapter 4

Bayesian Models for Sparse Signals

The close relation of redundant dictionaries, sparse representations, machine learning and statistics has given the incentive to seek solutions to the sparse recovery problem in a probabilistic setting. It turns out that the problem of sparse support recovery is very close to those of regression and classification. Usually the assumed statistical model is trained on the available dataset and a predictive distribution is constructed. More than often the predictor variables (model parameters) for a regression problem result being *sparse* or a large dataset can contain only a few classes of objects.

By employing Bayesian methods it is possible to formulate the Compressed Sensing problem into a regression problem and apply Bayesian inference methods to recover a sparse model for the given measurements. Actually most of the aforementioned sparse recovery principles - and as shall be shown, algorithms as well - have a probabilistic backbone. Jumping on the Bayesian bandwagon, the problem can be explored in a more meaningful way.

4.1 Maximum Likelihood and Maximum A-Posteriori Estimates

Consider the problem of reconstructing $\boldsymbol{x}$ from its noisy measurements $\boldsymbol{y}$ described by Equation (2.5), also shown here for convenience

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}. \tag{4.1}$$

This can be put in a probabilistic setting by assuming that the measurements follow some distribution. Following the justifiable norm, let us consider that a multivariate

normal distribution is employed. Then the probability distribution function for $\boldsymbol{y}$ given the measurements and noise variance is

$$p(\boldsymbol{y}|\boldsymbol{x},\sigma^2) = \mathcal{N}(\boldsymbol{H}\boldsymbol{x},\sigma^2\boldsymbol{I}) = \frac{1}{(2\pi\sigma^2)^{\frac{m}{2}}} e^{-\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2}$$

where $\boldsymbol{I}$ is the identity matrix. This expression is also referred to as the *likelihood* function of the data vector $\boldsymbol{y}$ given a set of model parameters $\boldsymbol{x}$.

The *Maximum Likelihood* (ML) estimate of $\boldsymbol{x}$ is found by minimising $-\log p(\boldsymbol{y}|\boldsymbol{x},\sigma^2)$ with respect to $\boldsymbol{x}$:

$$-\log p(\boldsymbol{y}|\boldsymbol{x},\sigma^2) = \frac{m}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2$$

which coincides with the *Least Squares* solution. It is straightforward that there is no trend towards a sparse $\boldsymbol{x}$ and it is a fact that the least-squares solution to an inverse problem is not a sparse one. In the statistician's dialect this is equivalent to saying that there is no preference towards sparse solutions expressed by the assumed model.

To remedy this problem - simply - a *prior* distribution is assumed for $\boldsymbol{x}$ which helps in expressing this need for a sparse solution, i.e. for the components x_i to have a tendency towards the zero value. For reasons that will become apparent, the Laplace distribution is chosen for this task. The probability density function of a Laplace distributed random variable x has the following form

$$f(x|\alpha,\beta) = \frac{1}{2\beta} e^{-\frac{|x-\alpha|}{\beta}}$$

and an example is shown in Figure 4.1. The mean value of x is given by α while its variance by $2\beta^2$.

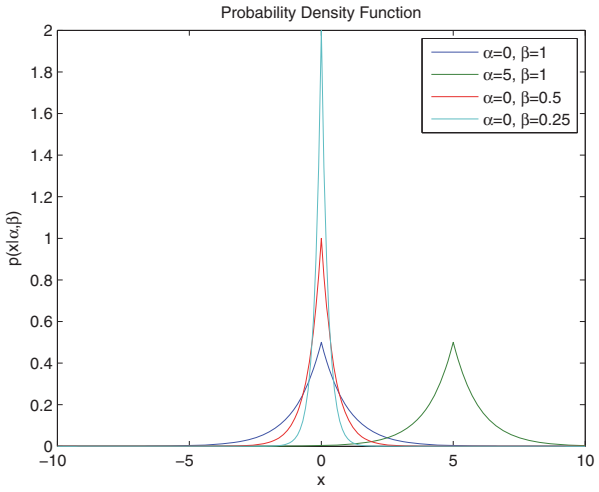


Figure 4.1: *Probability Density Function of the Laplace distribution.*

By assigning a Laplace prior over over x_i with $\alpha = 0$ and a suitably chosen β a prior belief can be expressed for x_i to attain smaller values. In which case the probability function becomes more peaked at zero. In a way β expresses how strong this belief is. Thus, since parameters x_i are independent from each other the prior distribution function becomes

$$p(\boldsymbol{x}|\beta) = \frac{1}{(2\beta)^m} e^{-\frac{\|\boldsymbol{x}\|_1}{\beta}}. \tag{4.2}$$

The prior is independent for each x_i but *hyper-parameter* β is shared.

According to the Bayes rule the posterior distribution for $\boldsymbol{x}$ is proportional to the product of the likelihood and the prior,

$$p(\boldsymbol{x}|\boldsymbol{y}, \sigma^2, \beta) \propto p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2) p(\boldsymbol{x}|\beta)$$

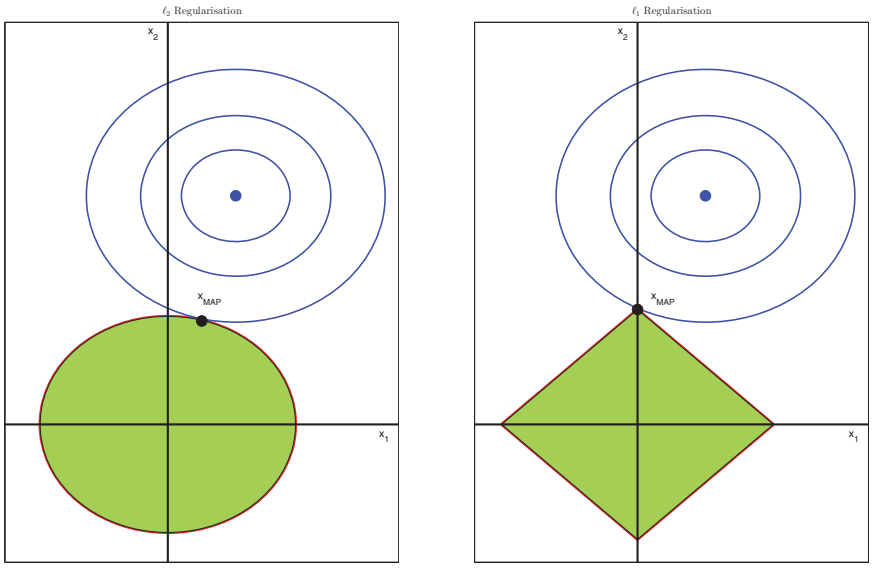
Finding the *Maximum a-posteriori* estimate (MAP) of $\boldsymbol{x}$ would require to maximise the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \sigma^2, \beta)$ or equivalently minimise $-\log p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2) - \log p(\boldsymbol{x}|\beta)$,

$$\boldsymbol{x}_{MAP} = \min_{\boldsymbol{x}} \left[C + \sigma^{-2} \left(\frac{1}{2} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + \frac{\sigma^2}{\beta} \|\boldsymbol{x}\|_1 \right) \right]$$

where C is a constant. This is the same as the finding the solution to the LASSO stated in Equation (2.7) for $\lambda = \frac{\sigma^2}{\beta}$ and BPDN in Equation (2.7). When $\sigma^2 \to 0$ then the MAP estimate for $\boldsymbol{x}$ under the Laplace prior is the same as the solution given by Basis Pursuit. The regularisation term λ in Equation (2.7) controls the trade-off between sparsity and quality of reconstruction. The MAP estimate can be seen as a regularisation of the ML.

In machine learning jargon, the use of prior distributions is said to avoid *over-fitting* which means that a model too complex is inferred. The regularisation term introduced by the prior helps in favouring *simple* models. Let us note that the Laplace distribution is not the only distribution fit for this task but other *exponential* super-Gaussian distributions can be used just like other $\ell_{0<p<1}$ norms can be used instead of the ℓ_1, resulting in different forms of regularisation (surely non-convex). These cases will not occupy us in this text since the techniques presented are not based solely on a specific form for the prior and can be applied. For example if a Gaussian distribution is used then the MAP would introduce a quadratic regularisation term leading to what is also known as *regularised least squares* or *Tikhonov regularisation*.

In Figure 4.2 two different choices of prior are shown, the Laplace and the Gaussian prior. Like in the case of ℓ_0 minimisation where replacing with the ℓ_1 norm makes the problem convex the same happens when adopting a different prior. More specifically by changing from the Gaussian prior to a super-Gaussian prior promotes



(a) *Gaussian prior* **(b)** *Laplace prior*

Figure 4.2: *Effect of regularisation on the sparsity of a solution in the case where $m = 2$. The blue concentric circles show the first part of the log-posterior which is the squared error term. The green area shows the constraint posed by the regularisation term. The MAP estimate is the point where these two areas meet.*

sparsity in the solutions.

4.1.1 Conjugate Prior Distributions

Employing various prior distributions like the Laplace prior comes with a price to pay, just like the requirement for sparse solutions. In simple terms by solving an under-determined system by least squares is a completely different story than requiring for that solution to be sparse.

The choice of prior has a significant impact on the tractability. For example when the prior is Gaussian $p(\boldsymbol{x}|\gamma) = \prod_{i=1}^{m} \mathcal{N}(0, \gamma)$ then the posterior $p(\boldsymbol{x}|\boldsymbol{y}, \sigma^2, \beta)$ admits a closed form solution which is also in the form of a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{y} \\ \boldsymbol{\Sigma} &= \left(\boldsymbol{H}^T\boldsymbol{H} + \sigma^2\gamma\boldsymbol{I}\right)^{-1}. \end{aligned}$$

In this case the choice of a Gaussian prior results in that the mean of the posterior distribution coincides with the MAP estimate $\boldsymbol{x}_{MAP} = \boldsymbol{\mu}$ (Figure 4.2a). Of course this does not hold for a Laplace prior and one has to solve the BPDN problem.

This convenient choice for a prior (even though it does not promote sparsity) where the posterior distribution belongs in the same family as the prior is called a *conjugate prior*. In this particular case; the exponential family of distributions. Quoting from a Wikipedia article, *a conjugate prior is an algebraic convenience,*

giving a closed-form expression for the posterior; otherwise a difficult numerical integration may be necessary. This means that an analytic expression for the posterior is possible depending on whether the prior is chosen wisely. The importance of this choice for a prior will become evident in the following discussions as well.

4.2 Sparse Bayesian Learning

The previous discussion on prior distributions might suggest that the probabilistic formulation of the sparse signal recovery problem has nothing to offer more than an intuitive and convenient representation. Fortunately and to our great advantage this is hardly the case since it is possible to construct models that give elegant solutions without the need to explicitly express a preference while still providing sparse solutions. By borrowing ideas from machine learning, it is possible to infer a sparse model for the data $\boldsymbol{y}$ with the help of a hierarchical prior model. This is achieved via an approximation process which allows a shortcut towards explicitly specifying a regularisation threshold. Even though this might seem to be computationally inefficient it is also proven that efficient algorithms exist. These algorithms will occupy us in a different chapter.

4.2.1 Graphical Models

In order to facilitate discussion and understanding a couple of paragraphs are spent to introduce the notion of Graphical models. A Graphical model is a way of graphically depicting the dependencies between the random variables that participate in a model. In such constructions it is easy to visualise a model and understand the computations that take place behind it for performing usual tasks such as inference (computation of posterior distributions of a set of variables) or even marginalisation (something related to *factor graphs*).

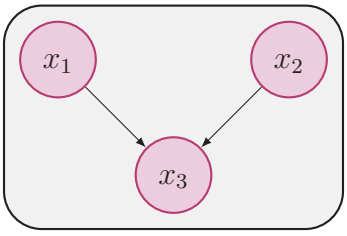


Figure 4.3: *A Simple Bayesian Network representing the relationship of random variables* x_1, x_2 *and* x_3.

In Figure 4.3 an example of a simple Bayesian network is shown. The graph represents the conditional dependencies between the three random variables x_1, x_2 and x_3. Every *node* represents a single or a group of variables while the *edges* indicate

a probabilistic relationship between the connected nodes. For this reason the edges are *directed*. Two unconnected nodes denote that the corresponding variables are conditionally independent. The graph in its entirety shows how the *joint distribution* can be factorised. For our little example this translates to:

$$p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2).$$

One can easily comprehend that variables x_1 and x_2 are independent when conditioned on x_3. A *Bayesian Network* is a type of graph like the above, bearing no cycles, i.e., no closed paths or a *Directed Acyclic Graph*.

In most of the cases the random variables associated are not all known and have to be estimated. Usually the input is described as the *data* or measurements like in the previous section. The model parameters that have to be estimated are usually called *latent* or *unobserved* and in our cases of interest they usually represent the sparse vector to be recovered. In the some of the models to follow the hyper-parameters of a model are also considered to be latent and have to be estimated. We will return to this at a different stage in this text. This section will not be occupied with the estimation algorithms *per se* but solely on the models and their meaning.

4.2.2 Sparse Bayesian Models

Following the discussion about graphical models and Bayesian networks we now introduce what is known in the wider bibliography as *Sparse Bayesian Models*. In most of the cases these are introduced as a *hierarchical* probabilistic model easily described as a graphical model. Probably the most notable of such models is the one introduced in [89] which is described below. The discussion to follow will be based on this original rendition. Then a smorgasbord of models is presented that employ various changes to achieve different results. This has the purpose to show how versatile and scalable the Bayesian methods are towards developing new algorithms and intuition.

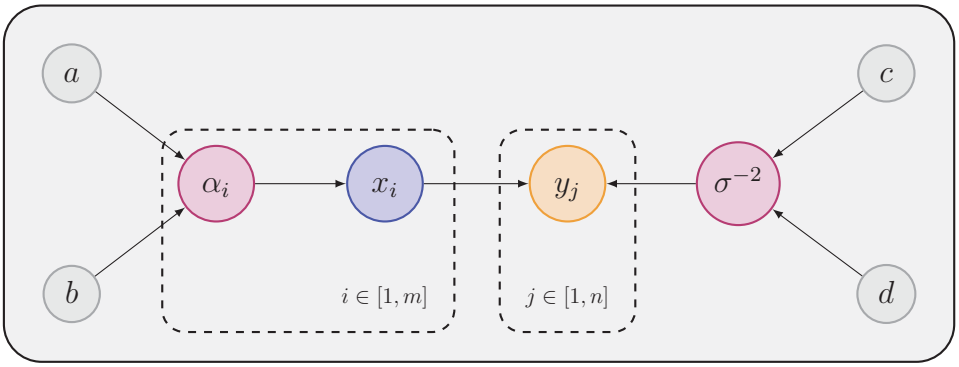


Figure 4.4: *Bayesian Network of the Relevance Vector Machine.*

In Figure 4.4 the aforementioned model is depicted [89]. A slight addition in the notation are the two dashed parts which are repeated m and n times respectively. The sparse parameter vector to be estimated is denoted as $\boldsymbol{x} \in \mathbb{R}^m$ and the measurements are gathered in vector $\boldsymbol{y} \in \mathbb{R}^n$. A hyper-prior distribution $p(\boldsymbol{\alpha})$ is placed above the parameter vector which takes as input another two parameters and the same happens for the noise variance. Values a, b, c and d denote the hyper-prior's parameters and for the moment it is assumed that they are not to be estimated and are deterministic constants. Note that the hyper-prior choice is specific for the model in [89] and other choices will be discussed later in the text.

This hierarchical model of distributions assigns a separate prior distribution to each component of the support x_i. At a first glance this might seem like a bad choice since it would lead to over-fitting (hence not a sparse model) due to the number of parameters to be estimated. Recall that in Section 4.1 there was only one set of m unknowns to estimate (the sparse vector) but in this case the number of unknowns is doubled with the addition of a hyper-prior distribution. The elegance of Bayesian methods shows that this is not the case. On the contrary it leads to efficient algorithms for sparse reconstruction. A central role plays the fact that the choice of the hyper-prior distribution is a conjugate distribution to the prior distribution.

More specifically it is assumed that the prior distribution for $\boldsymbol{x}$ is formulated as follows

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{i=1}^{m} \mathcal{N}(0, \alpha_i^{-1}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{A}^{-1}) \tag{4.3}$$

where *hyper-parameters* α_i control the inverse variance of the corresponding component x_i and matrix $\boldsymbol{A} = \text{diag}(\alpha_1, \cdots, \alpha_m)$. Doing a quick comparison with Equation (4.2) it is noticed that in this model a *separate* hyper-prior is placed over *each* component.

The hyper-prior distribution placed over $\boldsymbol{x}$ controls the variance of each x_i. In contrast to having a fixed prior like in Subsection 4.1 this allows for a consistent Bayesian treatment without introducing dependencies between variables x_i. This is key to realising sparsity in this framework. The hyper-prior is chosen to be

$$\begin{aligned} p(\boldsymbol{\alpha}) &= \prod_{i=0}^{m} \text{Gamma}(a, b) \\ p(\sigma^{-2}) &= \text{Gamma}(c, d), \end{aligned} \tag{4.4}$$

where $p(x) = \text{Gamma}(a, b) = \Gamma(a)^{-1} b^a x^{a-1} e^{-bx}$ is the Gamma distribution. The gamma distribution is chosen since it is a conjugate distribution to the precision (inverse variance) of the Gaussian distribution.

The parameters of this hyper-prior are chosen so that it is *uninformative*, i.e., so as to express no preference over any value for the corresponding x_i. A hyper-prior is also assigned to the noise variance of the measurement model. For the hyper-prior distributions to be uninformative the deterministic parameters a, b, c, d are set to zero or to near-zero values in practice. Note that such a prior is often described as *improper* because the posterior distribution cannot be normalised, hence it would not be an actual distribution. Essentially the true posteriors are approximated as if one used a proper hyper-prior with extreme values assigned to their parameters. In Figure 4.5 an example is given for the Gamma distribution for small values of the parameters.

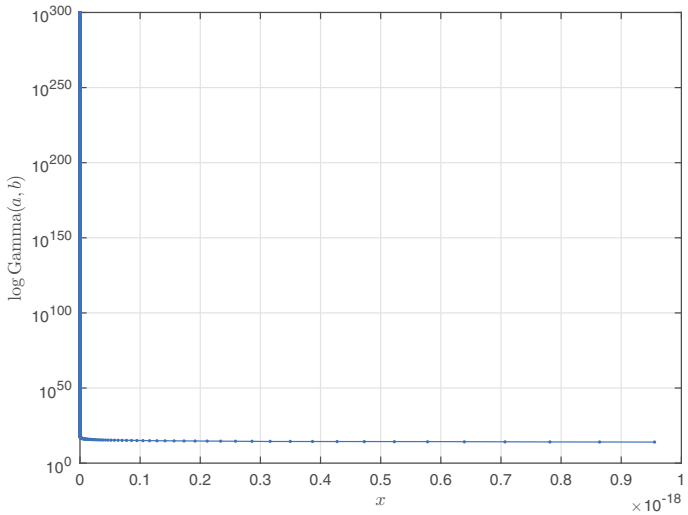


Figure 4.5: *Probability Density Function of Gamma distribution for parameter values* $a = b = 0.1^4$.

Bayesian Inference

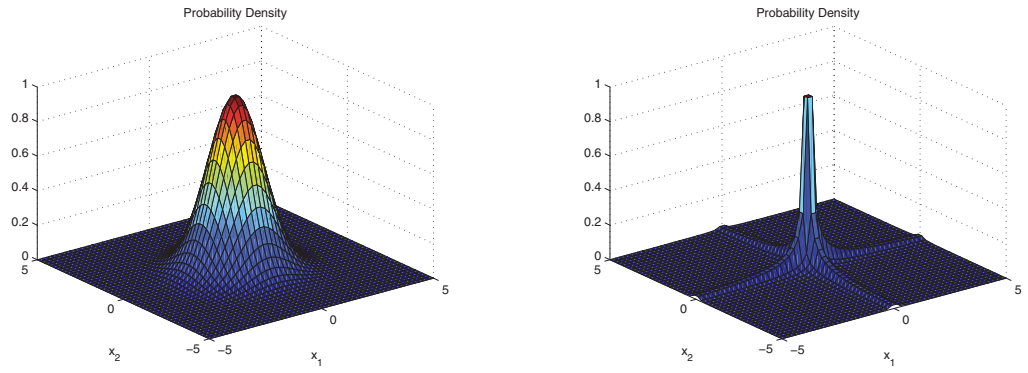
Writing down the joint distribution

$$p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2) = p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2)p(\boldsymbol{x}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\sigma^2).$$

and from the Bayes theorem, the posterior can be written as,

$$p(\boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2)}{p(\boldsymbol{y})}. \tag{4.5}$$

Any attempt to compute the expression above would come to a halt because of the term $p(\boldsymbol{y})$ being not possible to evaluate analytically in full [89]. We put a pause at this stage as far as this is concerned and for the remainder of the discussion it will be assumed that a suitable approximation exists. We will revert to this problem in more detail later in the text and inform the reader that the posterior is approximated



(a) $p(\boldsymbol{x}|\boldsymbol{\alpha})$ **(b)** $p(\boldsymbol{x})$

Figure 4.6: *(a) The probability distribution given the prior is a bivariate Normal. (b) By marginalising over the prior the resulting distribution is a* Student-t *distribution strongly peaked along the axes.*

via the *Expectation-Maximisation* algorithm or equivalently via *Type-II Maximum Likelihood* [9].

Promoting Sparsity

Based on the discussion on sparsity-promoting prior distributions in the previous section; a prior that places significant probability mass on near-zero values and exhibits heavy tails like the Laplace distribution is suitable for promoting a sparse solution. So far no such prior was shown since $p(x_i|\alpha_i)$ is a Gaussian while $p(\alpha_i)$ itself is a Gamma distribution. In order to work out the actual distribution of a single x_i from Equation (4.3) one must integrate over the hyper-prior α_i, i.e.

$$p(x_i) = \int p(x_i|\alpha_i)p(\alpha_i)d\alpha_i \propto \left(b + \frac{x_i^2}{2}\right)^{-\left(a+\frac{1}{2}\right)} = \frac{1}{|x_i|}$$

where in the last part of the equation the fact that $a = b = 0$ was considered. As it turns out the prior does manage to promote sparse solutions - $p(x_i) \propto \frac{1}{|x_i|}$ - because of its peak at zero, quite like the Laplace prior $p(x_i) \propto e^{-|x_i|}$. More specifically $p(\boldsymbol{x})$ follows a *Student-t* distribution. The graph in Figure 4.6 shows the comparison between $p(\boldsymbol{x}|\boldsymbol{\alpha})$ and $p(\boldsymbol{x})$ for $m = 2$, i.e. the bivariate case. It is evident that the prior over $p(\boldsymbol{x})$ shown on the right concentrates most of the probability mass on ridges along the axes as opposed to the Gaussian on the left.

4.2.3 Related Work

The authors of [3] - Babacan *et al.* - have attempted to make a good connection between $\ell_{0<p\leq 1}$ optimisation and Sparse Bayesian Learning by relating the models

discussed above with the following,

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + \tau\|\boldsymbol{x}\|_p^p$$

which is a slightly modified LASSO problem. For this the authors assign a Generalised Gaussian prior to the sparse vector which is very similar to the one shown above,

$$p(\boldsymbol{x}|\alpha) = C \cdot \alpha^{\frac{m}{p}} \left(-\alpha \sum_{i=1}^{m} |x_i|^p \right).$$

Note that the hyper-prior is *shared* among the model parameters and the authors follow a different approach to make the model more versatile.

Again, Babacan *et al.* in [4] have attempted the following hierarchical prior structure

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\gamma}) &= \prod_{i=1}^{m} \mathcal{N}(0, \gamma_i) \\ p(\gamma_i|\lambda) &= \text{Gamma}\left(1, \lambda/2\right) = \frac{\lambda}{2} e^{-\lambda\gamma_i/2} \\ p(\lambda|\nu) &= \text{Gamma}\left(\nu/2, \nu/2\right). \end{aligned}$$

The authors' goal with this model was to implement a Laplace prior over the parameters but the issue they faced was that the Laplace prior is not a conjugate prior to the precision of the Gaussian $p(y_i|\boldsymbol{h_i}^T\boldsymbol{x}, \sigma^2)$. By adopting the above three-stage hierarchical model the following marginal,

$$p(\boldsymbol{x}|\lambda) = \int p(\boldsymbol{x}|\boldsymbol{\gamma}) p(\boldsymbol{\gamma}|\lambda) d\boldsymbol{\gamma} = \frac{\lambda^{m/2}}{2^m} e^{-\sqrt{\lambda}\sum_{i=1}^{m}|x_i|},$$

indeed results in a Laplace prior over the sparse parameter vector. Let us stand for a moment and appreciate the elegance and versatility Bayesian models offer towards constructing new algorithms. In this model the value of λ has to be calculated also, being a random variable. This is accomplished via a *variational approach*, something which will be discussed in a separate chapter. To conclude with this form of prior a mention is made for [110] in which the authors have provided yet another prior for which the Laplace prior and the Gaussian-Gamma prior are special cases.

In another attempt to model the sparse coefficients of a system the authors in [21] use what is called an *algebraic-tailed prior* from the generalized Cauchy distribution. The authors base their intuition on the fact that these distributions better describe impulsive processes than the Gaussian distribution. The Generalised

Cauchy Distribution for a random variable z is defined as,

$$f(z) = p\frac{\Gamma(2/p)}{2(\Gamma(1/p))^2}\delta(\delta^p + |z|^p)^{-2/p},$$

where parameter δ controls the scale while p controls the tail. It is known to exhibit heavier tails than the Laplace distribution for $p = 1$ in which case it is called the *Meridian* distribution. Based on this the authors design the following prior,

$$p(\boldsymbol{x}|\delta) = \frac{\delta^m}{2^m}\prod_{i=1}^{m}\frac{1}{(\delta + |x_i|)^2}.$$

The MAP estimate when using the prior above becomes,

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + 2\sigma^2\|\boldsymbol{x}\|_{L,L_{1,\delta}}$$

where the norm,

$$\|\boldsymbol{u}\|_{L,L_{p,\delta}} = \sum_{i=1}^{m}\log\left(1 + \delta^{-p}|u_i|^p\right) \;\; \delta > 0.$$

The authors make their argument based on the fact that the $L, L_{p,\delta}$ quasi-norm can be used as an approximation to the ℓ_0 norm. One important feature which they quote is that this norm penalises a bit more mildly large deviations; hence being a bit more forgiving for impulsive processes.

The research team of Ji *et al.* in [84] have explicitly for the first time introduced to the community the definition of *Bayesian Compressive Sensing* which is the application of the model in [89] for the purpose of sparse signal recovery. A set of further minor modifications are also presented and the interested reader is redirected to the relevant bibliography. A year later members of the same team have went a step further to introduce the term *Multitask Compressive Sensing* in their paper [49]. In this work the authors consider the following problem,

$$\boldsymbol{y}_i = \boldsymbol{\Phi}_i\boldsymbol{x}_i + \boldsymbol{n}_i \;\; i \in [1, L] \;\; \boldsymbol{\Phi} \in \mathbb{R}^{n_i \times m}$$

where it is assumed that there exist L many sparse recovery problems, namely *tasks* and the tasks between them are not independent. Task i produces n_i many measurements and it is possible that each task contributes with a different number of measurements. The authors propose a hierarchical model to address this issue. This summary is restricted to the graphical model due to the length of the analysis. This demonstrates the other aspect of graphical models, the ease at which complex ideas can be communicated.

Using only intuition from the graphical model, there is enough flexibility to

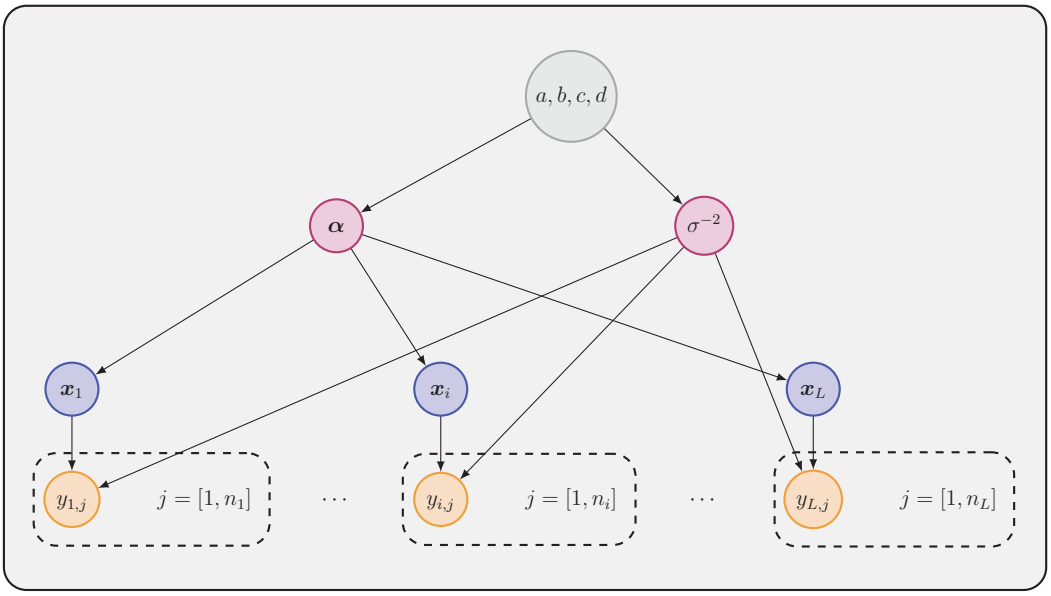


Figure 4.7: *Bayesian Network for Multi-task Compressive Sensing.*

model the individual tasks but also to capture the correlations between them. This is achieved by the shared hyper-prior over the sparse vectors from all the tasks $\boldsymbol{x}_i$. All the individual tasks contribute in inferring the hyper-prior vector $\boldsymbol{\alpha}$ which is then used to recover the individual $\boldsymbol{x}_i$.

A very inspired work by [94] showcases a different approach into constructing a prior. In short the authors propose the following hyper-prior,

$$p(\boldsymbol{\alpha}|\boldsymbol{\beta}) \propto e^{-c \cdot \text{Trace}(\sigma^2 \boldsymbol{H} \boldsymbol{\Sigma}_x \boldsymbol{H}^T)}$$

where $\boldsymbol{\Sigma}_x$ is the inferred covariance matrix of the posterior distribution. Without getting into much detail, the value in the exponent measures the degrees of freedom of the model. This is chosen as a regularising value of the sparsity along with control parameter c.

4.3 Conclusion

In this chapter the connection between sparse signal representations and Bayesian models was presented. In the beginning of this chapter we saw how the same problems can be formulated in the world of probabilities and how intuition is added in the recipe with the use of meaningful prior distributions. Most of the sparse recovery problems admit a probabilistic formulation. This formulation can then be extended and communicated in a very efficient manner by the use of Graphical models. The traditional recovery problems such as ℓ_1 optimisation can be recast, reformulated and improved by the use of a proper model that better fits the problem at hand.

The following list underscores some of the differences and similarities between more traditional sparse recovery methods and sparse Bayesian models.

- The first thing to notice is that instead of point estimates of the support $\boldsymbol{x}$, the Bayesian approach produces estimates of complete distributions. By computing the posterior $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2)$ statistical information on each of the components x_i is acquired. This is an appealing feature which will concern us in later chapters.

- Another appealing aspect which has been discussed, is the easiness at which these Bayesian models produce highly sparse solutions without any external tuning and sometimes they can be described as *automatic*. The hierarchical models that have been studied work in such a way so that they compute all the necessary parameters on their own. Recall that in deterministic methods which almost always require some sort of input such as the sparsity level or a regularisation parameter. Of course the performance of this Bayesian model depends on the model mismatch, i.e., whether the signal originated from the exact hierarchical model.

- One attribute against Bayesian methods is the difficulty to produce any provable performance guarantees like the ones for deterministic methods. Those theorems are probably the most valuable tool engineers have at their disposal to decide whether an algorithm is going to produce accurate results given a specific problem and noise levels. Unfortunately we do not have such luxuries with Bayesian methods due to the fact that most of them result in non-convex optimisation problems. In Chapter 5 to follow it will be demonstrated that a *different* analysis can lead to an alternative way of understanding how the specific hierarchical Bayesian model promotes sparsity. Via this analysis it is also shown that provable performance guarantees are indeed possible that are global. This bridges the gap between Bayesian methods and greedy pursuits for sparse signal recovery and is one of the contribution of this thesis.

Chapter 5

Bayesian Inference Algorithms for Sparse Recovery

Having discussed sparse signals from a Bayesian perspective let us point out that no *algorithm* has been discussed so far to actually perform *Bayesian Inference* on these sparse models. Inference is the mechanism of applying Bayes' theorem on the probabilistic model to compute the posterior probability of the unknown parameters in the model. Here a set of algorithms will be discussed that are able to carry out this task when a sparse Bayesian model is considered. The reasons why such an algorithm is needed is mainly attributed to the fact that the posterior usually does not admit a closed-form solution. The algorithm then implements a process of approximating the posterior. The algorithms come in two flavours; the *slow kind* which exhibit great computational and memory requirements and the *fast kind* which are the type used by the majority due to their efficiency. The reasons for this are presented below.

Researchers are also faced with a new dilemma. While traditional sparse recovery methods are amenable to rigorous mathematical analysis and performance guarantees, Bayesian inference algorithms have not been analysed to that degree. This is on one hand justifiable because of the context in which these algorithms have been developed in the first place (machine learning problems like regression and classification) but on the other hand their increased popularity for sparse signal recovery has left this gap related to their respective performance guarantees. There have been numerous attempts to "marry" the two, i.e., to put sparse Bayesian learning in a more theoretical foundation and point out how sparse recovery relates to the basic principles like ℓ_0 minimisation and basis pursuit.

More importantly there is a pronounced lack of such a theoretical foundation for the *fast kind* of Bayesian inference algorithms. In the later sections of this chapter a fine analysis is presented which points out a very fine connection between the fast inference algorithms and the greedy pursuit sparse recovery algorithms. The

benefits of such a connection are two-fold. Firstly it becomes possible to derive *global* performance guarantees based on metrics such as the mutual coherence of the matrix and the Restricted Isometry Property (Chapter 2). This is an important step since comparison with other recovery algorithms becomes possible something which was not possible before these innovations. Secondly, this uncovered relationship is found to extend previous results of other research teams that only provided *local* convergence guarantees. Lastly, great improvements are found to be possible by extending the Bayesian inference algorithms with concepts from traditional greedy pursuits. This innovation has a major impact on the range of applicable scenarios for Bayesian sparse recovery since fast inference algorithms with *superior* recovery performance can be constructed.

5.1 Type-II Maximum Likelihood

Let us start off this discussion with Equation (4.5) [89]

$$p(\boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{y}). \tag{5.1}$$

Looking at the above formula, the following integral is found to be impossible to compute analytically,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{x}, \boldsymbol{\alpha}, \sigma^2)d\boldsymbol{x}d\boldsymbol{\alpha}d\sigma^2.$$

Equation (5.1) is derived based on basic probability calculus. It is quickly recognised that this is the posterior over the parameters and is quite convenient since it is tractable and $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2)$ can be readily given in analytical form as a multivariate Gaussian distribution given by $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{y} \\ \boldsymbol{\Sigma} &= \left(\sigma^{-2}\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{A}\right)^{-1}. \end{aligned} \tag{5.2}$$

Focusing on the hyper-parameter posterior $p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{y})$; a suitable approximation is needed. Even though this still is an intractable integral, it is much easier to approximate. Taking the noise variance to be known, the easiest way out for an approximation is to adopt

$$p(\boldsymbol{\alpha}|\boldsymbol{y}) \approx \delta(\boldsymbol{\alpha}^*),$$

i.e., that the posterior is approximated with a delta distribution placed at the modes

(most probable values) of the actual posterior. Now, rewriting the posterior,

$$p(\boldsymbol{\alpha}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}),$$

one can see that under the *uninformative* prior assumption (see Chapter 4 and Figure 4.5) for $p(\boldsymbol{\alpha})$ that the most probable values $\boldsymbol{\alpha}^*$ can be found by optimising the likelihood $p(\boldsymbol{y}|\boldsymbol{\alpha})$. This approximation might seem overly too optimistic - one distribution collapsing to a delta distribution - but it proves to be highly practical and effective.

The procedure of optimising the quantity $p(\boldsymbol{y}|\boldsymbol{\alpha})$ is coined in the bibliography as *Evidence Approximation*, *Type-II Maximum Likelihood* or *the Evidence Procedure*. The following marginal log-likelihood is maximised with respect to $\boldsymbol{\alpha}$,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}) = \log p(\boldsymbol{y}|\boldsymbol{\alpha}) &= \log \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{\alpha})d\boldsymbol{x} = \log \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) \\ &= -\frac{1}{2}\left[n\log(2\pi) + \log|\boldsymbol{C}| + \boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y}\right] \end{aligned} \tag{5.3}$$

where $\boldsymbol{C} = \sigma^2\boldsymbol{I} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T$. By inspection it is evident that the above form is non-convex and optimisation cannot be performed straightforwardly. Optimising this cost function is the starting point for all inference algorithms to follow and subsequent analysis.

By using the determinant identity [75],

$$\begin{aligned}|\sigma^2\boldsymbol{I} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T| &= |\boldsymbol{A}|^{-1}|\sigma^2\boldsymbol{I}||\boldsymbol{A} + \sigma^{-2}\boldsymbol{H}^T\boldsymbol{H}| \\ \log|\boldsymbol{C}| &= -\log|\boldsymbol{\Sigma}| + n\log\sigma^2 - \log|\boldsymbol{A}|\end{aligned}$$

Using the Woodbury inversion identity [75],

$$\begin{aligned}\boldsymbol{C}^{-1} &= \sigma^{-2}\boldsymbol{I} - \sigma^{-2}\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T\sigma^{-2} \\ \boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y} &= \sigma^{-2}\boldsymbol{y}^T\left(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}\right) \\ &= \sigma^{-2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}\|^2 + \sigma^{-2}\boldsymbol{y}^T\boldsymbol{H}\boldsymbol{\mu} - \sigma^{-2}\boldsymbol{\mu}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{\mu} \\ &= \sigma^{-2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}\|_2 + \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \sigma^{-2}\boldsymbol{\mu}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{\mu} \\ &= \sigma^{-2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}\|_2 + \boldsymbol{\mu}^T\boldsymbol{A}\boldsymbol{\mu}.\end{aligned}$$

Some insight is gained by observing that the result of the above relates to a log-likelihood function computed at the mean of the posterior of Equation (5.2). Each component contributes as much as the prior $\boldsymbol{\alpha}$ will allow via the diagonal matrix $\boldsymbol{A}$.

Combining the two,

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\left[n\log\sigma^2 - \log|\boldsymbol{\Sigma}| - \log|\boldsymbol{A}| + \sigma^{-2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}\|_2 + \boldsymbol{\mu}^T\boldsymbol{A}\boldsymbol{\mu}\right] \tag{5.4}$$

The expression above resembles the log-likelihood function computed at the mean of the posterior of the parameters while being regularised by the corresponding variances on the main diagonal of matrix $\boldsymbol{A}$. Each component is found to contribute only as much as its variance will allow.

5.1.1 Direct Optimisation

Quoting [89], Equation (5.4) can be directly optimised with respect to $\log \alpha_i$,

$$\frac{\partial \mathcal{L}}{\partial \log \alpha_i} = \frac{1}{2}\left[1 - \alpha_i(\mu_i^2 + \Sigma_{ii})\right] = 0 \tag{5.5}$$

The optimal value for the hyper-parameters is acquired,

$$\alpha_i^* = \frac{1}{\mu_i^2 + \Sigma_{ii}}. \tag{5.6}$$

The noise variance also according [89] is computed in a similar manner,

$$\sigma_*^2 = \frac{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2}{n - \sum_i (1 - \alpha_i \Sigma_{ii})}$$

but in this text it will be assumed to be known to keep the discussion clutter-free.

5.1.2 Expectation-Maximisation

The Expectation-Maximisation (EM) is a general technique for computing maximum likelihood solutions for models that have what is known as *latent* variables. In this case the model parameters $\boldsymbol{x}$ are considered to be the latent variables. Without plunging into much details about the derivation of the algorithm the results are collocated for our model of interest.

The algorithm iterates between two steps, namely the Expectation step and the Maximisation stem. At the *E-step* it is assumed that the hyper-parameters have been estimated somehow and are fixed and the posterior over the model parameters is computed by Equation (5.2).

For the *M-step* the following log-likelihood is optimised with respect to α_i,

$$\mathrm{E}_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha},\sigma^2}\left[\log p(\boldsymbol{x}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})\right]$$

which give the update rule for the hyper-parameters

$$\alpha_i^* = \frac{1}{\langle x_i^2 \rangle} = \frac{1}{\mu_i^2 + \Sigma_{ii}}$$

where $\langle x_i^2 \rangle = \mathrm{E}_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}\sigma^2}[x_i^2] = \mu_i^2 + \Sigma_{ii}$. This is a direct result for the Gaussian distribution.

The hyper-parameter update rules are the same for when using direct optimisation of the likelihood function above. The noise variance expression ends up being slightly different and the interested reader is redirected to [89].

5.1.3 Computational Complexity

The iterative algorithm then proceeds by repeated application of Equations (5.2) and the resulting optimal hyper-parameter expressions. Both of these methods even though highly practical and straightforward have high computational requirements in the order of $\mathcal{O}(m^3)$ which is the cost for computing the inverse of the variance matrix of the posterior of the parameters. This is indeed the case - at least for the initial iterations of the algorithm - since a sparse parameter vector is expected. It is expected that most of the hyper-parameter values will converge to near-infinity values (i.e. near-zero variance for the corresponding model parameters) thus a mechanism can be constructed to exclude these parameters from the model. Basically if α_i^* is found to be greater than some high threshold τ or close to the machine precision then the corresponding columns of $\boldsymbol{\Sigma}$ can be excluded from its re-estimation thus reducing the cost for the inverse. Even then, the cost would be prohibitive for massive datasets until the algorithm had begun to converge. The authors of [90] have proposed a more elegant way of keeping complexity under control without sacrificing inference performance. This is discussed in the next section.

5.2 Fast Marginal Likelihood Maximisation

In [37, 90] a different *fast* approach is introduced and analysed which suggests a highly efficient algorithm for the iterative maximisation of Equation (5.3). The key difference with the previously proposed algorithms is that the special form of the cost function (5.3) allows for a convenient re-writing

$$\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}(\boldsymbol{\alpha}_{-i}) + \ell(\alpha_i) \tag{5.7}$$

where subscript $-i$ means the exclusion of the corresponding hyper-parameter from the calculations. The second term on the left is the remainder and bears dependence only on the single hyper-parameter. By differentiating with respect to α_i and equating to zero the analytical expressions for a single α_i are found.

The authors apply some basic linear algebra to separate the cost function in two distinct parts to achieve separation. The relevant matrix is basically rewritten as a

sum of rank-1 updates,

$$\begin{aligned}\boldsymbol{C} &= \left(\sigma^2\boldsymbol{I}_n + \sum_{j\neq i}\alpha_j^{-1}\boldsymbol{h}_j\boldsymbol{h}_j^T\right) + \alpha_i^{-1}\boldsymbol{h}_i\boldsymbol{h}_i^T \\ &= \boldsymbol{C}_{-i} + \alpha_i^{-1}\boldsymbol{h}_i\boldsymbol{h}_i^T\end{aligned}$$

where the subscripted quantities indicate computation with removal of the corresponding column. Using the determinant identity and the Woodbury inversion lemma,

$$\begin{aligned}|\boldsymbol{C}| &= |\boldsymbol{C}_{-i}||1 + \alpha_i^{-1}\boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{h}_i| \\ \boldsymbol{C}^{-1} &= \boldsymbol{C}_{-i}^{-1} - \frac{\boldsymbol{C}_{-i}^{-1}\boldsymbol{h}_i\boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{h}_i}.\end{aligned}$$

Combining all of the above, Equation (5.3) can be rewritten,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}) &= \frac{1}{2}\Bigg[n\log(2\pi) + \log|\boldsymbol{C}_{-i}| + \boldsymbol{y}^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{y} \\ &\quad - \log(\alpha_i) + \log(\alpha_i + \boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{h}_i) - \frac{\left(\boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{y}\right)^2}{\alpha_i + \boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{h}_i}\Bigg] \\ &= \mathcal{L}(\boldsymbol{\alpha}_{-i}) + \frac{1}{2}\left[\log(\alpha_i) - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right] \\ &= \mathcal{L}(\boldsymbol{\alpha}_{-i}) + \ell(\alpha_i).\end{aligned}$$

The following very useful quantities have been defined

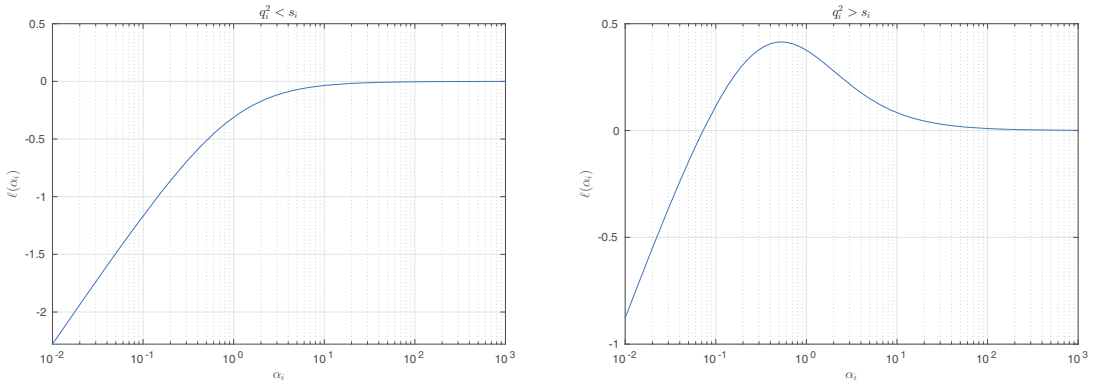
$$s_i = \boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{h}_i \quad q_i = \boldsymbol{h}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{y}. \tag{5.8}$$

The next step is to directly optimise function $\mathcal{L}(\boldsymbol{\alpha})$ which is now a lot easier due to this explicit separation. Optimisation of $\ell(\alpha_i)$ with respect to α_i gives two distinct and mutually exclusive stationary points,

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i}, \qquad \text{for } \; q_i^2 > s_i \tag{5.9}$$

$$\alpha_i = +\infty, \qquad \text{for } \; q_i^2 \leq s_i. \tag{5.10}$$

Equations (5.10), (5.9) suggest an analytic *pruning* rule, i.e., the means to explicitly exclude a parameter from the model based on a criterion on q_i and s_i. This is something which the EM algorithm or the direct Type-II ML do not provide. This has a tremendous effect on the complexity since for a sparse support the computational effort will decrease as most of the α_i will satisfy (5.10). When $\alpha_i = +\infty$ then



(a) $q_i^2 < s_i$ **(b)** $q_i^2 > s_i$

Figure 5.1: *(a) The log-likelihood function $\ell(\alpha_i)$ for the case when the corresponding parameter is found to be irrelevant to the dataset since the variance tends to zero ($\alpha_i \to +\infty$). (b) The opposite case for when the log-likelihood exhibits a finite maximum and the corresponding parameter has to be kept.*

this automatically means that the most probable value for the corresponding $x_i = 0$. A graphical example of the pruning rule is shown in Figure 5.1. On the left-hand side the case where the value of α_i diverges can be seen, hence the maximum value is $+\infty$. On the right-hand side the opposite case is shown where the maximum value attained is positive and finite.

This finding suggests an iterative algorithm which takes advantage of this analytic pruning. Of course this involves a "greedy" step since there has to be a schedule on which basis functions are removed, added or altered. The steps of the iterative algorithm developed in [90] are given in Algorithm 4.

Implementation Aspects

The algorithm outlined in Algorithm 4 iterates through all the columns of $\boldsymbol{H}$ and applies the pruning rule

$$\theta_i = q_i^2 - s_i. \tag{5.11}$$

At the same time it calculates the increase in the likelihood for the specific change suggested by the value of θ_i. The change which causes the likelihood to increase the most is applied and the parameters of the posterior $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2)$ are updated. This alternating process repeats until the change in the likelihood $\mathcal{L}(\boldsymbol{\alpha})$ falls below some threshold. This procedure indicates a greedy approach in which the change to achieve the greater increase to the log-likelihood will be applied.

The authors also propose an efficient way to compute the necessary quantities in Equation (5.8). They propose that complete records are kept in vectors $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{q}}$. In the case that $0 < \alpha_i < +\infty$ then the following records are kept

$$\hat{s}_i = \boldsymbol{h}_i^T \boldsymbol{C}^{-1} \boldsymbol{h}_i \quad \hat{q}_i = \boldsymbol{h}_i^T \boldsymbol{C}^{-1} \boldsymbol{y}$$

Algorithm 4 Fast Marginal Likelihood Maximisation

Input: $\boldsymbol{H}$, $\boldsymbol{y}$

Initialise:

1. Initialise σ^2 to some appropriate value.
2. $\mathcal{T} = \{$ index i for which α_i is minimum $\}$.
3. Compute $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ for $\mathcal{T}$.

Iteration:

1. For each $i \in [1, m]$:
   - Compute $\theta_i = q_i^2 - s_i$.
   - Calculate the increase $\Delta\ell_i(\alpha_i)$ for all the possible changes in $\mathcal{T}$:
     - $i \in \mathcal{T}$ but $\theta_i > 0$ and i should *remain* in $\mathcal{T}$.
     - $i \notin \mathcal{T}$ but $\theta_i > 0$ and i should be *added* to $\mathcal{T}$.
     - $i \in \mathcal{T}$ but $\theta_i \leq 0$ and i should be *removed* from $\mathcal{T}$.
2. Select index i for which $\Delta\ell_i$ is maximised and apply the corresponding change (addition, re-estimation, removal).
3. Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the new $\mathcal{T}$ and α_i.
4. If the change in $\mathcal{L}(\boldsymbol{\alpha})$ is below some threshold then quit.

Output:

1. Estimated support set $\mathcal{T}$ and sparse support $\boldsymbol{x} = \boldsymbol{\mu}$ with covariance matrix $\boldsymbol{\Sigma}$.

and the aforementioned quantities are calculated as

$$s_i = \frac{\alpha_i \hat{s}_i}{\alpha_i - \hat{s}_i} \quad q_i = \frac{\alpha_i \hat{q}_i}{\alpha_i - \hat{q}_i}.$$

In the case where $\alpha_i \to +\infty$, then the Woodbury inversion lemma is used

$$\hat{s}_i = \sigma^{-2}\boldsymbol{h}_i^T\boldsymbol{h}_i - \sigma^{-4}\boldsymbol{h}_i^T\boldsymbol{H}_{\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{h}_m \tag{5.12}$$

$$\hat{q}_i = \sigma^{-2}\boldsymbol{h}_i^T\boldsymbol{y} - \sigma^{-4}\boldsymbol{h}_i^T\boldsymbol{H}_{\mathcal{I}}\boldsymbol{\Sigma}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{y}. \tag{5.13}$$

Subset $\mathcal{I} = \{i : 0 < \alpha_i < +\infty\}$ denotes those indices for which the corresponding hyper-parameters escape the pruning rule, i.e., are kept in the model. The sub-scripted quantities indicate calculation based only on the indices in $\mathcal{L}$; which is a sub-matrix $\boldsymbol{H}_{\mathcal{I}}$ and the covariance $\boldsymbol{\Sigma}_{\mathcal{I}} = \left(\sigma^{-2}\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{H}_{\mathcal{I}} + \boldsymbol{A}_{\mathcal{I}}\right)^{-1}$.

Notice the immense decrease in computational requirements since from $\mathcal{O}(m^3)$ we fall down to $\mathcal{O}(k^3)$ where $k \ll m$ is the level of sparsity in the signal. The quantities in Equation (5.13) are far easier to compute than the inverse of the full covariance matrix.

In the original work [90] the authors give update formulae to update the param-

eters $\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$ and the increase in $\mathcal{L}$ for each type of change in the support set $\mathcal{T}$. This way complexity is further kept to reasonable levels by avoiding complete matrix inversions at each iteration. The update formulae comes after some straightforward but tedious mathematical manipulations so they are not given here. A complete list can be found in [90].

Comparing Basis Pursuit De-noising and Fast Marginal Likelihood Maximisation one notices that the latter is almost completely *automatic*. There is no need to priorly assume a specific sparsity level k for the measured signal. The analytic pruning rule resulting from the evidence approximation manages to infer the sparsity level from the data itself without the need for an external regularising parameter.

The original algorithms presented in [89] empirically have negligible differences when compared to FMLM. Of course the difference in computational requirements is huge, especially for large datasets. The trade-off this time comes in the form of greediness. The FMLM as presented in Algorithm 4 is a greedy algorithm, unlike its original computationally hungry cousins. At each time instant it makes a greedy choice based on the greatest increase in the likelihood function. In later chapters it is demonstrated how this leaves a lot of room for improvement.

5.3 Evaluation of Sparse Bayesian Learning

With each algorithm must come a set of reassuring statements on how the algorithm performs. Such statements were made in the form of those excellent theorems regarding sparse signal recovery with convex relaxation and greedy pursuits. These theorems usually come in two flavours; the first one having to do with the existence and uniqueness of a solution (this part usually is independent of the process, i.e. the algorithm) and the second one having to do about how close a given algorithm can get to the best solution.

A huge amount of effort has been made by researchers towards proving such conditions about the models that have been discussed above. We stand mostly on the work of David. P. Wipf and Bashkar D. Rao whose research team has made many contributions into building an framework about how sparse recovery is exactly related with the hierarchical models for promoting sparsity. Their work has helped researchers understand a great deal about the relationship between ℓ_0 minimisation, Basis Pursuit and the regularising effect of a great class of regularising prior distributions.

5.3.1 Sparse Bayesian Learning and ℓ_0-norm minimisation

The first attempt to relate the two was in [105]. Among other results the authors in [107, 106] provide a very useful theorem regarding the Type-II ML cost function - or Sparse Bayesian Learning (SBL) - in Equation (5.3) and ℓ_0 norm minimisation.

Theorem 8 ([106]). *Let $\mathcal{X}_0$ denote the set of vectors that globally minimise the well known problem,*

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \;\; \textit{so that} \;\; \boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}.$$

Also let,

$$\mathcal{X}(\sigma^2) = \left\{ \boldsymbol{x}^* : \boldsymbol{x}^* = (\boldsymbol{H}^T\boldsymbol{H} + \sigma^2\boldsymbol{A})^{-1}\boldsymbol{H}^T\boldsymbol{y},\; \boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}) \right\}.$$

In the limit of $\sigma^2 \to 0$, if $\boldsymbol{x} \in \mathcal{X}(\sigma^2)$ then $\boldsymbol{x} \in \mathcal{X}_0$.

The theorem puts forward a clear declaration that optimising the SBL cost function is in essence an alternative path towards approximating ℓ_0 norm minimisation. The result is pretty much self explanatory; if a solution minimises the SBL cost function in the noiseless setting then that solution is also one of the optimal solutions to the ℓ_0 problem.

In the noisy case the authors provide yet another theorem regarding local minima (i.e. suboptimal solutions).

Theorem 9 ([106]). *Every local minimum of $\mathcal{L}$ is achieved at a sparse solution, regardless of the noise level.*

Theorem 9 basically states that at all cases the SBL will return a sparse vector no matter what. This proves the highly practical features of this model. In line with Theorem 8 it seems that optimising $\mathcal{L}$ is a very good choice for performing ℓ_0 optimisation since globally optimal solutions can be achieved in the absence of noise, whereas in any other case one would recover a sparse vector.

5.3.2 Sparse Bayesian Learning and Convex Relaxation

The next question the authors have provided the answer to is whether the SBL cost function is actually a better choice than traditional convex relaxation with ℓ_p regularisation. In [103, 108]the authors have shown the exact relationship between these two strategies and have shown exactly why it is better to choose SBL.

Theorem 10 ([103, 108]). *Consider the prior $p(x_i) = \int \mathcal{N}(0, \alpha_i^{-1}) p(\alpha_i) d\alpha_i \propto 1/|x_i|$*

and the following cost function

$$\mathcal{K}(\boldsymbol{\alpha}) = \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y} + \sum_{i=1}^{m} \log \alpha_i$$

with $\boldsymbol{\alpha} \geq \mathbf{0}$ *and* $\boldsymbol{C} = \sigma^2 \boldsymbol{I} + \boldsymbol{H} \boldsymbol{A}^{-1} \boldsymbol{H}^T$. *Then* $\boldsymbol{\alpha}$ *is the global minimum of* $\mathcal{K}$ *iff*

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}^{-1} \boldsymbol{H}^T \left(\sigma^2 \boldsymbol{I} + \boldsymbol{H} \boldsymbol{A}^{-1} \boldsymbol{H}^T \right)^{-1} \boldsymbol{y} \\ \boldsymbol{A} &= \operatorname{diag}(\boldsymbol{\alpha}) \end{aligned}$$

is a global minimum of

$$\mathcal{M}(\boldsymbol{x}) = -2 \log p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + \sigma^2 \sum_{i=1}^{m} |x_i|.$$

This correspondence extends to local minima as well.

It is useful to note that the following holds true,

$$\begin{aligned} p(x_i) &= \int p(x_i|\alpha_i) p(\alpha_i) d\alpha_i \\ &= \frac{{b_i}^{a_i} \Gamma(a_i + 1/2)}{\sqrt{2\pi}\Gamma(a_i)} (b_i + {x_i^2}/2)^{-(a_i + 1/2)} \propto \frac{1}{|x_i|} \end{aligned}$$

where in the last step it was assumed that $a_i = b_i = 0$.

By comparing Equation (5.3) with the results from Theorem 10 we conclude that MAP estimation of $\boldsymbol{x}$ (cost function $\mathcal{M}$ in the theorem) is actually a limiting case of the Type-II method with the $\log|\boldsymbol{C}|$ term missing from the former. So one can adopt a similar optimisation form to Type-II ML over the hyper-parameter space and the corresponding computed value of the posterior mean $\boldsymbol{x}$ would be the same as performing MAP over the parameter space. Basically this shows that Type-II methods are more general.

Theorem 11 ([103, 108])**.** *Consider cost function*

$$\mathcal{M}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + \sigma^2 g(\boldsymbol{x})$$

with penalty function

$$g(\boldsymbol{x}) = \min_{\boldsymbol{\alpha} \geq \mathbf{0}} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + \log|\boldsymbol{C}|. \tag{5.14}$$

Then

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}^{-1} \boldsymbol{H}^T \left(\sigma^2 \boldsymbol{I} + \boldsymbol{H} \boldsymbol{A}^{-1} \boldsymbol{H}^T \right)^{-1} \boldsymbol{y} \\ \boldsymbol{A} &= \operatorname{diag}(\boldsymbol{\alpha}) \end{aligned}$$

is a global minimum of $\mathcal{M}(\boldsymbol{x})$ *iff* $\boldsymbol{\alpha}$ *is a global minimum of* $\mathcal{L}(\boldsymbol{\alpha})$ *in Equation* (5.3). *This correspondence extends to local minima as well.*

Therefore the opposite has been proven as well, that Type-II methods can be seen as problems similar to finding the MAP estimate. The theorem basically shows a way of obtaining the Type-II posterior mode directly in the parameter space.

5.3.3 The SBL Cost Function Local Minima

The authors give rigorous proof that the SBL cost function is superior over Type-I methods in that they exhibit far less local minima. In [82] certain conditions are given regarding the worst-case scenario for a sparse signal that can be given as input to SBL and they show how it compares with other methods. Later in [109, 108] the authors investigated the effect of the prior distributions on the local minima. A toy problem is presented which is derived from the bibliography to showcase one important aspect of SBL.

Consider the case where $\boldsymbol{H}^{10\times 11}$ such that the null-space of this matrix is consisted of one vector. Any solution of $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ can be written as $\boldsymbol{x} = \boldsymbol{x}_0 + a\boldsymbol{v}$ where $\boldsymbol{v}$ belongs in the null-space of $\boldsymbol{H}$, since $\boldsymbol{H}(\boldsymbol{x}_0 + a\boldsymbol{v}) = \boldsymbol{y}$ with $\boldsymbol{x}_0$ being the maximally sparse solution. Using this small experiment one can plot any function over any $\boldsymbol{x}$ with respect to the real multiplier a, i.e. there is only one global minimiser which is found when $a = 0$. This can be used to compare an ℓ_p regularisation term with the Type-II penalty in Equation (5.14).

In Figure 5.2 such a comparison is made between the traditional prior used in SBL and ℓ_p for $p = 0.01$. It is easy enough to see that there are indeed 11 possible local minima for the ℓ_p while those have been "smoothed-out" as the authors say in the Type-II. Based on this intuition the Type-II methods process less local minima hence there as little chance that the algorithm will provide one as a solution.

5.4 Improved Fast Marginal Likelihood Maximisation

In the previous section a series of theoretical results on the relationship of the Type-II cost function have been presented. Those results have solidified the fact that indeed Sparse Bayesian Learning provides a Bayesian short-cut towards ℓ_0 optimisation and a good one for that matter. The authors have even provided the necessary conditions under which the aforementioned cost function exhibits less local minima and the worst-case scenarios, i.e., the cases at which SBL is likely to perform badly.

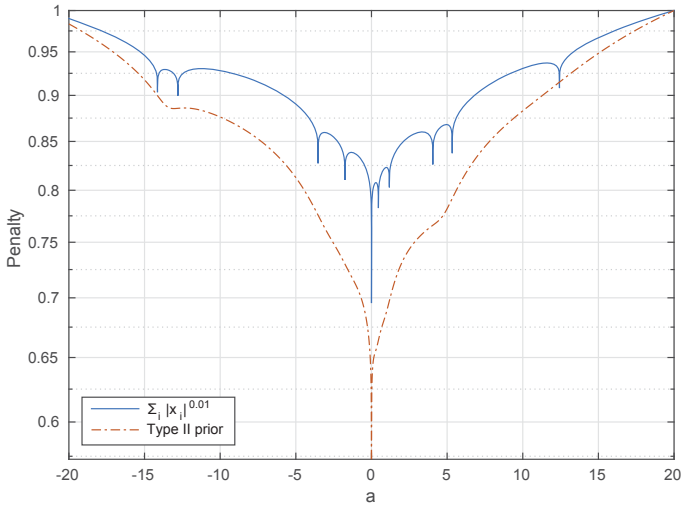


Figure 5.2: *Comparison of local minima between Type-II ML prior and ℓ_p regularisation with $p = 0.01$.*

In this section focus is turned on the set of greedy techniques for performing inference, namely the Fast Marginal Likelihood Maximisation (FMLM) [90] algorithm. The reader should take a quick glance at Algorithm 4 to notice that the FMLM has a greedy behaviour. The criterion on which it bases its decisions is the value of the log-likelihood. This resembles other greedy pursuits albeit using a criterion usually related to the residual error.

Given the fact that greedy pursuits exhibit reduced computational and memory requirements than other sparse recovery principles an attempt has been put forward into combining the two. One of the major contributions of this thesis is the derivation of the direct relationship of FMLM and other greedy algorithms such as the Orthogonal Matching Pursuit and the Subspace Pursuit that have been introduced in the initial chapters. The result is a set of superior sparse Bayesian inference algorithms that exhibit far better qualities.

Another important contribution of this thesis is the derivation of provable performance guarantees for FMLM. These performance guarantees have been presented for the first time to the author's knowledge in [56, 53]. These include the sufficient conditions for exact sparse signal recovery based on the mutual coherence and the Restricted Isometry Property.

5.4.1 A Properly Scaled FMLM algorithm

A usual practice towards drawing conclusions regarding performance and algorithm behaviour is to focus on how FMLM behaves in the limit of zero noise variance, $\sigma^2 \to 0$. This was also the case with [107, 109]. The results are then extrapolated to the noisy-case. Usually under the zero noise assumption the probabilistic models collapse into deterministic ones. This makes it difficult to perform analysis for hi-

erarchical models in which proper probability distributions play a significant role in inference.

Theorem 12 (Karseras, Dai)**.** *For any given hyper-parameter vector $\boldsymbol{\alpha}$, define the set*

$$\mathcal{I} \triangleq \{1 \leq i \leq m :\ 0 < \alpha_i < \infty\}.$$

Then

$$\lim_{\sigma^2 \to 0} \sigma^2 \mathcal{L} = \left\| \boldsymbol{y} - \boldsymbol{H}_{\mathcal{I}} \boldsymbol{H}_{\mathcal{I}}^{\dagger} \boldsymbol{y} \right\|_2^2,$$

where $\boldsymbol{H}_{\mathcal{I}}$ is the sub-matrix of $\boldsymbol{H}$ formed by the columns indexed by $\mathcal{I}$, and $\boldsymbol{H}_{\mathcal{I}}^{\dagger}$ denotes the pseudo-inverse of $\boldsymbol{H}_{\mathcal{I}}$. In particular, if $\boldsymbol{y} \in \operatorname{span}(\boldsymbol{H}_{\mathcal{I}})$, then

$$\lim_{\sigma^2 \to 0} \sigma^2 \mathcal{L} = 0.$$

If $|\mathcal{I}| < n$, then the unscaled cost function behaves as

$$\lim_{\sigma^2 \to 0} \mathcal{L} = -\infty.$$

Proof. Consider any given hyper-parameter vector $\boldsymbol{\alpha}$. Assume the set of indices $\mathcal{I} = \{1 \leq i \leq m :\ 0 < \alpha_i < \infty\}$ for which the corresponding entries of $\boldsymbol{\alpha}$ have positive and finite values. Also let $\mathcal{D} = [1, m] - \mathcal{I}$ denote the remaining indices. For vector $\boldsymbol{\alpha}$ matrix $\boldsymbol{C}$ can be written as,

$$\begin{aligned} \boldsymbol{C} &= \sigma^2 \boldsymbol{I} + \boldsymbol{H}_{\mathcal{I}} \boldsymbol{A}_{\mathcal{I}}^{-1} \boldsymbol{H}_{\mathcal{I}}^T + \boldsymbol{H}_{\mathcal{D}} \boldsymbol{A}_{\mathcal{D}}^{-1} \boldsymbol{H}_{\mathcal{D}}^T \\ &= \sigma^2 \boldsymbol{I} + \boldsymbol{H}_{\mathcal{I}} \boldsymbol{A}_{\mathcal{I}}^{-1} \boldsymbol{H}_{\mathcal{I}}^T \end{aligned}$$

since $\alpha_i = +\infty,\ \forall\ i \in \mathcal{D}$. Subscripts $\mathcal{I}, \mathcal{D}$ denote the sub-matrices formed by the corresponding subsets.

In order to derive the properly scaled version of the cost function, the determinant and the inverse of matrix $\boldsymbol{C}$ are rewritten as follows

$$\begin{aligned} \log |\boldsymbol{C}| &= -n \log \left| \sigma^{-2} \boldsymbol{I} \right| + \log \left| \boldsymbol{I} + \sigma^{-2} \boldsymbol{H}_{\mathcal{I}} \boldsymbol{A}_{\mathcal{I}}^{-1} \boldsymbol{H}_{\mathcal{I}}^T \right| \\ \boldsymbol{C}^{-1} &= \sigma^{-2} \boldsymbol{I} - \sigma^{-2} \boldsymbol{H}_{\mathcal{I}} \left(\sigma^2 \boldsymbol{A}_{\mathcal{I}} - \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{H}_{\mathcal{I}} \right)^{-1} \boldsymbol{H}_{\mathcal{I}}^T \end{aligned}$$

where the matrix inversion lemma [44] was used in the derivation of the second equation. Now the cost function becomes

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}) &= -n \log \sigma^{-2} + \log \left| \boldsymbol{I} + \sigma^{-2} \boldsymbol{h}_{\mathcal{I}} \boldsymbol{A}_{\mathcal{I}}^{-1} \boldsymbol{H}_{\mathcal{I}}^T \right| + \\ &\quad \sigma^{-2} \boldsymbol{y}^T \left(\boldsymbol{y} - \boldsymbol{H}_{\mathcal{I}} \left(\sigma^2 \boldsymbol{A}_{\mathcal{I}} + \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{H}_{\mathcal{I}} \right)^{-1} \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{y} \right) \\ &= o(\sigma^{-2}) + \sigma^{-2} \boldsymbol{y}^T \left(\boldsymbol{y} - \boldsymbol{H}_{\mathcal{I}} \left(\sigma^2 \boldsymbol{A}_{\mathcal{I}} + \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{H}_{\mathcal{I}} \right)^{-1} \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{y} \right). \end{aligned}$$

In the case where noise variance approaches zero,

$$\begin{aligned}\lim_{\sigma^2\to 0}\sigma^2\mathcal{L}(\boldsymbol{\alpha}) &= \boldsymbol{y}^T\left(\boldsymbol{y}-\boldsymbol{H}_{\mathcal{I}}\left(\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{H}_{\mathcal{I}}\right)^{-1}\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{y}\right)\\ &= \boldsymbol{y}^T\left(\boldsymbol{y}-\boldsymbol{H}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^{\dagger}\boldsymbol{y}\right)\end{aligned} \tag{5.15}$$

where in the last step the expression for the pseudo-inverse of a tall matrix is used. Now let $\boldsymbol{y}_p = \boldsymbol{H}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^{\dagger}\boldsymbol{y}$ denote the projection of $\boldsymbol{y}$ on the span of $\boldsymbol{H}_{\mathcal{I}}$ and $\boldsymbol{y}_r = \boldsymbol{y}-\boldsymbol{y}_p$ the corresponding residual. It holds that

$$\langle \boldsymbol{y}, \boldsymbol{y}_r\rangle = \langle \boldsymbol{y}_p+\boldsymbol{y}_r, \boldsymbol{y}_r\rangle = \|\boldsymbol{y}_r\|_2^2,$$

since,

$$\begin{aligned}\langle \boldsymbol{y}_p, \boldsymbol{y}_r\rangle &= \left(\boldsymbol{H}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^{\dagger}\boldsymbol{y}\right)^T\left(\boldsymbol{y}-\boldsymbol{H}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^{\dagger}\boldsymbol{y}\right)\\ &= \left(\boldsymbol{H}_{\mathcal{I}}^{\dagger}\boldsymbol{y}\right)^T\left(\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{y}-\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{H}_{\mathcal{I}}\left(\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{H}_{\mathcal{I}}\right)^{-1}\boldsymbol{H}_{\mathcal{I}}^T\boldsymbol{y}\right) = 0.\end{aligned}$$

From the above, Equation (5.15) becomes

$$\lim_{\sigma^2\to 0}\sigma^2\mathcal{L}(\boldsymbol{\alpha}) = \|\boldsymbol{y}_r\|_2^2.$$

Let $\mathcal{K}$ be the correct support set for vector $\boldsymbol{x}$. Further assume that $\boldsymbol{y} \in \text{span}(\boldsymbol{H}_{\mathcal{I}})$, then naturally $\mathcal{K}\subseteq\mathcal{I}$. It then follows that

$$\begin{aligned}\boldsymbol{y}_r &= \boldsymbol{y}-\boldsymbol{H}_{\mathcal{I}}\boldsymbol{H}_{\mathcal{I}}^{\dagger}\boldsymbol{y}\\ &= \boldsymbol{y}-\boldsymbol{H}_{\mathcal{K}}\boldsymbol{H}_{\mathcal{K}}^{\dagger}\boldsymbol{y} = \boldsymbol{0}\end{aligned}$$

and subsequently

$$\lim_{\sigma^2\to 0}\sigma^2\mathcal{L}(\boldsymbol{\alpha}) = 0.$$

which completes the first part of the proof.

For the second part, consider the unscaled cost function in the noiseless setting. It suffices to see that the determinant term becomes

$$|\boldsymbol{C}| = \left|\boldsymbol{H}_{\mathcal{I}}\boldsymbol{A}_{\mathcal{I}}^{-1}\boldsymbol{H}_{\mathcal{I}}^T\right| = 0$$

since $\text{rank}(\boldsymbol{H}_{\mathcal{I}}\boldsymbol{A}_{\mathcal{I}}^{-1}\boldsymbol{H}_{\mathcal{I}}^T) < n$ and $\boldsymbol{C}\in\mathbb{R}^{n\times n}$. It then follows that

$$\lim_{\sigma^2\to 0}\mathcal{L}(\boldsymbol{\alpha}) = -\infty$$

which completes the proof for Theorem 12. □

Theorem 12 suggests that in the case where the noise variance approaches zero the problem becomes equivalent to recovering a subset $\mathcal{I}$ with a minimal number of elements. This would correspond to the optimal solution, minimising the scaled cost function. It is actually the same principle governing many sparse recovery algorithms including the OMP [91] and the SP [26]. In the case of zero noise variance minimisation of the unscaled cost function can be achieved by any subset $\mathcal{I}$ for which the corresponding hyper-parameters take positive and finite values. The scenarios analysed in [107] are special cases of Theorem 12.

The scaling affects certain parts of the fast inference algorithm to optimise the cost function. The cost function has a unique maximum with respect to a single hyper-parameter and two cases exist

$$\alpha_i = \begin{cases} \frac{s_i^2}{\theta_i} & \text{if } \theta_i > 0, \\ +\infty & \text{if } \theta_i \leq 0, \end{cases}$$

where $\theta_i = q_i^2 - s_i$ while $s_i = \boldsymbol{h}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{h}_i$ and $q_i = \boldsymbol{h}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{y}$.

Interest is turned towards the case where $\theta_i > 0$, i.e. basis function i is selected. Let us first define $\mathcal{I}$ to be the set of indices i for which $0 < \alpha_i < +\infty$. Also let $\mathcal{D} = \mathcal{I} - i$ be the set formed by the removal of element i from index set $\mathcal{I}$. Then the following reformed quantities are derived,

$$\sigma^{-2}\boldsymbol{\Sigma} = \left(\sigma^2 \boldsymbol{A}_{\mathcal{I}} + \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{H}_{\mathcal{I}}\right)^{-1} \tag{5.16}$$

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{y},$$

$$\sigma^2 \boldsymbol{C}_{-i}^{-1} = \boldsymbol{I} - \boldsymbol{H}_{\mathcal{D}}(\sigma^2 \boldsymbol{A}_{\mathcal{D}} + \boldsymbol{H}_{\mathcal{D}}^T \boldsymbol{H}_{\mathcal{D}})\boldsymbol{H}_{\mathcal{D}}^T, \tag{5.17}$$

$$\bar{s}_i = \sigma^2 s_i = \boldsymbol{h}_i^T \left(\sigma^2 \boldsymbol{C}_{-i}^{-1}\right) \boldsymbol{h}_i$$

$$\bar{q}_i = \sigma^2 q_i = \boldsymbol{h}_i^T \left(\sigma^2 \boldsymbol{C}_{-i}^{-1}\right) \boldsymbol{y}.$$

Subsequently the expression for the optimal α_i given all other $\alpha_j, j \neq i$, becomes

$$\alpha_i = \frac{\sigma^4 s_i^2}{\sigma^4 q_i^2 - \sigma^4 s_i} = \frac{\bar{s}_i^2}{\theta_i}$$

where now $\theta_i = \bar{q}_i^2 - \sigma^2 \bar{s}_i$. When $\sigma^2 \to 0$ this expression becomes

$$\begin{aligned} \alpha_i &= \frac{\left(\boldsymbol{h}_i^T (\boldsymbol{h}_i - \boldsymbol{H}_{\mathcal{D}} \boldsymbol{H}_{\mathcal{D}}^{\dagger} \boldsymbol{h}_i)\right)^2}{\left(\boldsymbol{h}_i^T (\boldsymbol{y} - \boldsymbol{H}_{\mathcal{D}} \boldsymbol{H}_{\mathcal{D}}^{\dagger} \boldsymbol{y})\right)^2 + \sigma^2 \left(\boldsymbol{h}_i^T (\boldsymbol{h}_i - \boldsymbol{H}_{\mathcal{D}} \boldsymbol{H}_{\mathcal{D}}^{\dagger} \boldsymbol{h}_i)\right)} \\ &= \frac{(\boldsymbol{h}_i^T \boldsymbol{h}_{i,r})^2}{(\boldsymbol{h}_i^T \boldsymbol{y}_{i,r})^2} = \frac{\bar{s}_i^2}{\bar{q}_i^2} \end{aligned} \tag{5.18}$$

where $\boldsymbol{h}_{i,r}$ denotes the residual vector from the projection of $\boldsymbol{h}_i$ on the span of

$\boldsymbol{H}_{\mathcal{D}}$. The same holds for the residual $\boldsymbol{y}_{i,r}$. In the derivation of Equation (5.18) the following fact was also used

$$\lim_{\sigma^2 \to 0} \sigma^2 \boldsymbol{C}_{-i}^{-1} = \boldsymbol{I} - \boldsymbol{H}_{\mathcal{D}} \boldsymbol{H}_{\mathcal{D}}^{\dagger}.$$

The importance of deriving these expressions can be shown with a simple example. If $\sigma^2 = 0$ is substituted in Equation (5.2) to compute the posterior mean then it is easily verified that performance degrades due to the inversion of a badly conditioned covariance matrix. There seems to be a gap one has to jump when altering between the noisy and the noiseless setting. However, adopting the properly scaled reformed quantities it is possible to draw on further results.

Performance Guarantees

By studying the inference algorithm from a non-Bayesian standpoint we notice that basically there is a measure by which all the possible atoms of the dictionary are judged to participate in the model. In the original rendition of the FMLM in [90] the authors provide several choices. One of them being at random, i.e, choose a basis function i and apply the corresponding change which can be either an *addition*, a *removal* or a *re-estimation* of the affected quantities including the posterior mean and variance. The most popular choice that the authors propose is to greedily choose and apply that modification which causes the cost function $\mathcal{L}(\boldsymbol{\alpha})$ to increase the most. Thus the criterion becomes the difference $\Delta\mathcal{L}_i(\alpha_i)$.

On the other hand, algorithms such as the OMP and SP make decisions based on different criteria. For example the OMP makes greedy choices based on the correlation values while the SP employs additional steps which effectively allow for a more relaxed selection regime. A very good question arises; whether by adopting a different selection strategy in FMLM it is possible to achieve better performance guarantees for sparse reconstruction and if there is any room for improvement. The following theorem sheds some light on this matter.

Theorem 13 (Karseras, Dai)**.** *Assume the noiseless setting* $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ *where* $\boldsymbol{H} \in \mathbb{R}^{n \times m}$ *and* $\boldsymbol{h}_i^T \boldsymbol{h}_i = 1$ *for all* $1 \leq i \leq m$*. Further assume that* $h = \max \left| \boldsymbol{h}_i^T \boldsymbol{h}_j \right|$ *for* $1 \leq i \neq j \leq m$*. A variant of the FMLM algorithm based on one of the following selection criteria:*

1. *the maximum value of* $\sigma^2 \Delta\mathcal{L}_i$
2. *the maximum value of* x_i
3. *the minimum value of* α_i*.*

is equivalent to FMLM and recovers all k-sparse signals exactly given the sufficient condition,

$$h < \frac{0.375}{k}.$$

A variant of the FMLM algorithm based on the maximum value of $\theta_i = \bar{q}_i$*, recovers all k-sparse signals exactly given the sufficient condition*

$$h < \frac{0.5}{k}.$$

Proof. The mutual coherence h for a matrix $\boldsymbol{H} \in \mathbb{R}^{n \times m}$ is defined as

$$h = \max_{1 \leq i \neq j \leq m} |\boldsymbol{h_i}^T \boldsymbol{h_j}|$$

assuming that $\boldsymbol{h}_i^T \boldsymbol{h}_i = 1,\ \forall i \in [1, m]$.

Assume that the selection criterion for the basis vectors is the value of the corresponding hyper-parameter as give by Equation (5.18),

$$\alpha_i = \frac{\bar{s}_i^2}{\bar{q}_i^2} = \frac{(\boldsymbol{h}_i^T \boldsymbol{h}_{i,r})^2}{(\boldsymbol{h}_i^T \boldsymbol{y}_{i,r})^2}. \tag{5.19}$$

Let $\mathcal{I}$ denote the true support set for $\boldsymbol{x}$ with $\mathcal{I} = k$ and $\mathcal{D} = [1, m] - \mathcal{I}$ its complement. Further consider j^* to be the index for the minimum α in the correct support set,

$$j^* = \arg\min_j \alpha_j \text{ so that } j^* \in \mathcal{I}.$$

Towards proving the sufficient condition for α_i the following requirement must hold,

$$|\alpha_{j^*}| < |\alpha_{i \in \mathcal{D}}|. \tag{5.20}$$

In order to proceed the left-hand side must be bounded from above while the right-hand side must be bounded from below. The numerator of Equation (5.19) is bounded,

$$1 - kh \leq |\boldsymbol{h_i}^T \boldsymbol{h_{i,r}}| \leq 1. \tag{5.21}$$

The denominator of the left-hand side of Inequality (5.20) can be bounded as follows

$$\begin{aligned} |\boldsymbol{h}_{j^*}^T \boldsymbol{y}| &\geq |x_{j^*}| - \left| \sum_{i \neq j^*, i \in \mathcal{I}} x_i \boldsymbol{h}_i^T \boldsymbol{h}_{j^*} \right| \geq |x_{j^*}| - h \sum_{i \neq j^*, i \in \mathcal{I}} |x_i| \\ &\geq |x_{j^*}| - h\sqrt{k}\|\boldsymbol{x}\|_2 \geq \left(\frac{1}{\sqrt{k}} - h\sqrt{k} \right) \|\boldsymbol{x}\|_2. \end{aligned} \tag{5.22}$$

where the fact that $\|\boldsymbol{x}\|_\infty \geq \frac{\|\boldsymbol{x}\|_2}{\sqrt{k}}$ and $\|\boldsymbol{x}\|_1 \leq \sqrt{k}\|\boldsymbol{x}\|_2$ were used.

Moving to the right-hand side of Inequality (5.20), for any $i \in \mathcal{D}$ the following holds

$$\begin{aligned} |\boldsymbol{h}_i^T \boldsymbol{y}| &\leq \left| \sum_{j \in \mathcal{I}} x_j \boldsymbol{h}_i^T \boldsymbol{h}_j \right| \leq \sum_{j \in \mathcal{I}} |x_j| |\boldsymbol{h}_i^T \boldsymbol{h}_j|, \\ &= \sum_{j \in \mathcal{I}} |x_j| h \leq h\sqrt{k} \|\boldsymbol{x}\|_2. \end{aligned} \tag{5.23}$$

By applying both the bounds from (5.22), (5.23) and (5.21)

$$\begin{aligned} \alpha_{i \in \mathcal{D}} &\geq \frac{1 - kh}{h\sqrt{k}\|\boldsymbol{x}\|_2} \\ \alpha_{j^*} &\leq \frac{1}{\left(\frac{1}{\sqrt{k}} - h\sqrt{k}\right) \|\boldsymbol{x}\|_2}. \end{aligned}$$

In order to derive the sufficient condition for exact recovery of every k-sparse signal the above bounds are substituted in the requirement posed by Inequality (5.20) and this gives

$$\begin{aligned} \frac{1 - kh}{h\sqrt{k}\|\boldsymbol{x}\|_2} &> \frac{1}{\left(\frac{1}{\sqrt{k}} - h\sqrt{k}\right) \|\boldsymbol{x}\|_2} \\ k^2h^2 - 3kh + 1 &< 0 \end{aligned}$$

By solving the above inequality one finally arrives at

$$h < \frac{3 - \sqrt{5}}{2k} \approx \frac{0.375}{k}. \tag{5.24}$$

This concludes the proof for the sufficient condition for when the minimum value of α_i is used to select the basis functions. Theorem 13 suggests that when the maximum value of x_i or the maximum value of $\sigma^2 \Delta\mathcal{L}_i$ are used as the selection rule the sufficient condition for the mutual coherence of $\boldsymbol{H}$ is the same. To prove this we refer to [90] for the formula for a single component x_i,

$$x_i = \frac{\boldsymbol{h}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{y}}{\alpha_i + \boldsymbol{h}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{h}_i} = \frac{\boldsymbol{h}_i^T \left(\sigma^2 \boldsymbol{C}_{-i}^{-1}\right) \boldsymbol{y}}{\sigma^2 \alpha_i + \boldsymbol{h}_i^T \left(\sigma^2 \boldsymbol{C}_{-i}^{-1}\right) \boldsymbol{h}_i}$$

which in the limit of zero noise,

$$x_i = \lim_{\sigma^2 \to 0} = \frac{\boldsymbol{h}_i^T \left(\sigma^2 \boldsymbol{C}_{-i}^{-1}\right) \boldsymbol{y}}{\sigma^2 \alpha_i + \boldsymbol{h}_i^T \left(\sigma^2 \boldsymbol{C}_{-i}^{-1}\right) \boldsymbol{h}_i} = \frac{\bar{q}_i}{\bar{s}_i^2}.$$

Likewise by taking the limit for the expression of $\sigma^2\Delta\mathcal{L}$ one gets,

$$\sigma^2\Delta\mathcal{L}_i = \frac{\bar{q}_i}{\bar{s}_i}.$$

The expressions above for x_i and $\sigma^2\Delta\mathcal{L}_i$ are very similar to the expression for α_i given in Equation (5.19) and that the analysis for deriving the sufficient condition is performed in the same way. Finally one arrives at the same condition given in (5.24) for all three criteria. This concludes the proof for the first part of the theorem.

Moving to the second part of the theorem where the selection criterion is

$$\theta_i = \bar{q}_i = \boldsymbol{h}_i^T \boldsymbol{y}_{i,r}.$$

By applying the same analysis as for α_i which in fact is slightly easier due to the absence of a numerator one arrives at the conclusion that

$$h < \frac{0.5}{k}.$$

This completes the proof for Theorem 13.

□

Theorem 13 gives some very useful insight. Actually now we become more certain about how different basis selection criteria perform. If one replaces $\Delta\mathcal{L}_i$ as the selection criterion in FMLM then different performance is attained. Basically the same sufficient condition for exact recovery is required for when choosing any of the three criteria involved in the theorem. More importantly this condition is relaxed when choosing based on the value of θ_i. This is an important result since under the Bayesian framework anything like this would be meaningless, yet from what we know about greedy pursuits this is perfectly normal. Even then, it is still possible to keep updating the mean and the variance of the posterior like in the un-altered FMLM algorithm.

FMLM Variants

Here the theoretical results are combined to form a new flavour of the FMLM algorithm which is another novelty presented in this thesis. Changing the selection criterion should result in better performance. Below a set of algorithms is presented as variants of the FMLM. In the last part of this section a far superior algorithm is presented that not only does it use a different criterion but a renovated selection strategy altogether based on the Subspace Pursuit. It is also shown how the new algorithm achieves sparse recovery with far more relaxed conditions.

Algorithm 5 FMLM-$\mathcal{X}$

Input: $\boldsymbol{H}, \boldsymbol{y}, \sigma^2$
Initialise:
- $\hat{\mathcal{T}} = \{\text{index } i \in [1, m] \text{ for maximum } |\boldsymbol{h}_i^T \boldsymbol{y}|\}$.
Iteration:
- Calculate values of α_i and criterion $\mathcal{X}$ for $i \in [1, m] \setminus \hat{\mathcal{T}}$.
- $\mathcal{T}' = \hat{\mathcal{T}} \cup \{$ index i corresponding to the best value for $\mathcal{X}$ for $i \notin \hat{\mathcal{T}}$.
- Calculate values α_i for $i \in \mathcal{T}'$.
- $\tilde{\mathcal{T}} = \{i \in \mathcal{T}' : 0 < \alpha_i < +\infty\}$.
- If $|\bar{\mathcal{L}}_{\tilde{\mathcal{T}}} - \bar{\mathcal{L}}_{\hat{\mathcal{T}}}| <$ tol. then compute $\sigma^{-2}\boldsymbol{\Sigma}, \boldsymbol{\mu}$ for $\tilde{\mathcal{T}}$ and quit. Otherwise set $\hat{\mathcal{T}} = \tilde{\mathcal{T}}$ and continue.
Output:
- Estimated support set $\tilde{\mathcal{T}}$ and sparse signal $\tilde{\boldsymbol{x}}$ with $\tilde{\boldsymbol{x}}_{\tilde{\mathcal{T}}} = \boldsymbol{\mu}$ and $\tilde{\boldsymbol{x}}_{\tilde{\mathcal{T}}^c} = \boldsymbol{0}$, where $\tilde{\mathcal{T}}^c = \{[1, m] - \tilde{\mathcal{T}}\}$.
- Estimated covariance matrix $\sigma^{-2}\boldsymbol{\Sigma}$.

Theorem 13 gives the incentive to redesign the inference algorithm. More specifically the FMLM algorithm can be reassembled to admit OMP-like performance guarantees based on different criteria as the first part of Theorem 13 suggests. Actually the inference algorithm greatly resembles the OMP; where the basis functions are recovered sequentially with decreasing order of correlation with the residual signal. As the second part of Theorem 13 suggests, performance guarantees equivalent to those of the OMP are achievable.

This relationship with the OMP becomes more evident by observing Equation (5.18). In the noiseless case and if θ_i is used as the selection criterion in FMLM instead of α_i, then this variant is the same algorithm as the OMP.

In Algorithm 5 these variants are presented. Set $\mathcal{X} = \{x_i, \theta_i, \alpha_i, \sigma^2 \Delta\mathcal{L}_i\}$ denotes the different choices in criteria so FMLM-$\mathcal{X}$ means the variant based on one of the available criteria suggested by 13. Note that a tolerance threshold is used to assess whether the cost function has stabilised in the 6^{th} step of the algorithm.

5.4.2 Bayesian Subspace Pursuit

Having gained consciousness of how the inference algorithm behaves, it is now possible to alter the selection strategy completely to improve the algorithm in terms of the exact sparse signal recovery. Following this rationale, further progress can be made by adopting more sophisticated selection rules.

Results from [26] are the motivation to extend the FMLM-$\mathcal{X}$ algorithm to a less greedy optimisation procedure by borrowing ideas from the Subspace Pursuit (SP) algorithm. The SP selects a subset of basis functions at each time instant based on correlation maximisation, but adds a backtracking step so as to retain only the sparse components with the largest magnitudes. The proposed algorithm,

Algorithm 6 Bayesian Subspace Pursuit

Input: $\boldsymbol{H}, \boldsymbol{y}, \sigma^2$
Initialise:
- $\hat{\mathcal{T}} = \{\text{index } i \in [1, m] \text{ for minimum } \alpha_i = \frac{1}{|\boldsymbol{h}_i^T \boldsymbol{y}|}\}$.
Iteration:
- Store $\alpha_{max} = \arg\max_{\alpha_i} |\alpha_i|$ for $i \in \hat{\mathcal{T}}$.
- Calculate values α_i and $\theta_i = \bar{q}_i^2 - \bar{s}_i$ for $i \in [1, m]$.
- Calculate values $l_{\theta_i>0} = |\{i \in [1, m] : \theta_i > 0\}|$ and $l_{\alpha_i \leq \alpha_{max}} = |\{i \in [1, m] : |\alpha_i| \leq \alpha_{max}\}|$.
- If $l_{\theta_i>0} = 0$, then $k = l_{\alpha_i \leq \alpha_{max}} + 1$ else $k = l_{\theta_i>0} + l_{\alpha_i < \alpha_{max}}$.
- $\mathcal{T}' = \hat{\mathcal{T}} \cup \{\text{indices corresponding to } k \text{ smallest values of } \alpha_i \text{ for } i \in [1, m]\}$.
- Compute $\sigma^{-2}\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ for $\mathcal{T}'$.
- $\tilde{\mathcal{T}} = \{\text{indices corresponding to } k \text{ largest non-zero values of } \boldsymbol{\mu} \text{ for which } 0 < \alpha_i < +\infty\}$.
- If $|\bar{\mathcal{L}}_{\tilde{\mathcal{T}}} - \bar{\mathcal{L}}_{\hat{\mathcal{T}}}| <$ tol. then quit. Otherwise set $\hat{\mathcal{T}} = \tilde{\mathcal{T}}$ and continue.
Output:
- Estimated support set $\tilde{\mathcal{T}}$ and sparse signal $\tilde{\boldsymbol{x}}$ with $\tilde{\boldsymbol{x}}_{\tilde{\mathcal{T}}} = \boldsymbol{\mu}$ and $\tilde{\boldsymbol{x}}_{\tilde{\mathcal{T}}^c} = \boldsymbol{0}$, where $\tilde{\mathcal{T}}^c = \{[1, m] - \tilde{\mathcal{T}}\}$.
- Estimated covariance matrix $\sigma^{-2}\boldsymbol{\Sigma}$.

termed here Bayesian Subspace Pursuit (BSP), shares the Bayesian background of SBL while the basis selection part is improved by the SP core. The algorithm still remains agnostic of the sparsity level and still provides useful statistical information to use in the tracking steps. The redesigned algorithm is presented in Algorithm 6 while for comparison the reader can revert to Algorithm 2 in Chapter 2.

Performance Guarantees

Theorem 14 (Karseras, Dai)**.** *Given the measurements $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ where $\boldsymbol{H} \in \mathbb{R}^{n \times m}$, Algorithm 6 recovers all k-sparse signals $\boldsymbol{x}$ exactly if matrix $\boldsymbol{H}$ satisfies the Restricted Isometry Property (RIP) with parameter*

$$\delta_{3k} < 0.205.$$

Proof. To facilitate the analysis of the theorem we rely on the following widely used lemmas. Only the main results are shown here whilst a complete proof can be found in [26, 17].

Lemma 2. *The RIP constant δ_k is a monotonically increasing function of k, i.e.,*

$$\delta_k \leq \delta_{k'}$$

for any two integers $k \leq k'$.

Lemma 3. *Assume* $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ *and subsets* $\mathcal{I}, \mathcal{J} \subset [1, m]$ *with* $\mathcal{I} \cap \mathcal{J} = \emptyset$ *and* $\delta_{|\mathcal{I}|+|\mathcal{J}|} < 1$. *Then it holds that*

$$\left\| \boldsymbol{H}_{\mathcal{I}}^T \boldsymbol{H}_{\mathcal{J}} \boldsymbol{a} \right\|_2 \leq \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\boldsymbol{a}\|_2$$

for any vector $\boldsymbol{a} \in \mathbb{R}^{|\mathcal{J}|}$.

Lemma 4. *Assume* $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ *and subsets* $\mathcal{I}, \mathcal{J} \subset [1, m]$ *with* $\mathcal{I} \cap \mathcal{J} = \emptyset$ *and* $\delta_{|\mathcal{I}|+|\mathcal{J}|} < 1$. *Also let* $\boldsymbol{y} \in \text{span}(\boldsymbol{H}_{\mathcal{I}})$. *The following holds:*

$$\frac{1 - 2\delta_{|\mathcal{I}|+|\mathcal{J}|}}{1 - \delta_{|\mathcal{I}|+|\mathcal{J}|}} \|\boldsymbol{y}\|_2 \leq \|\boldsymbol{y} - \boldsymbol{H}_{\mathcal{J}} \boldsymbol{H}_{\mathcal{J}}^{\dagger} \boldsymbol{y}\|_2$$

Assuming the same setting as earlier, consider $\mathcal{T}$ to be the correct support set for sparse signal $\boldsymbol{x}$. Furthermore, consider sets $\tilde{\mathcal{T}}$, $\hat{\mathcal{T}}$ and $\mathcal{T}'$ as defined in Algorithm 6. We cite the following two theorems:

[26, Th.3]: It holds that

$$\|\boldsymbol{x}_{\mathcal{T}-\mathcal{T}'}\|_2 \leq \frac{\sqrt{10\delta_{2k}}}{1 + \delta_{2k}} \|\boldsymbol{x}_{\mathcal{T}-\hat{\mathcal{T}}}\|_2$$

[26, Th.4]: It holds that

$$\|\boldsymbol{x}_{\mathcal{T}-\tilde{\mathcal{T}}}\|_2 \leq \frac{1 + \delta_{3k}}{1 - \delta_{3k}} \|\boldsymbol{x}_{\mathcal{T}-\mathcal{T}'}\|_2$$

These two theorems help in establishing the relationship between the steps of Algorithm 6 as far as reconstruction error is concerned.

More specifically in order for the algorithm to recover all k-sparse signals exactly the following must hold:

$$\|\tilde{\boldsymbol{y}}_r\|_2 < \|\hat{\boldsymbol{y}}_r\|_2 \tag{5.25}$$

where $\tilde{\boldsymbol{y}}_r = \boldsymbol{y} - \boldsymbol{H}_{\tilde{\mathcal{T}}} \boldsymbol{H}_{\tilde{\mathcal{T}}}^{\dagger} \boldsymbol{y}$ and $\hat{\boldsymbol{y}}_r = \boldsymbol{y} - \boldsymbol{H}_{\hat{\mathcal{T}}} \boldsymbol{H}_{\hat{\mathcal{T}}}^{\dagger} \boldsymbol{y}$. In order to make the connection between these two quantities and derive the sufficient condition we make use of Lemmas 2 and 3. Specifically:

$$\begin{aligned}
\|\tilde{\boldsymbol{y}}_r\|_2 &= \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\tilde{\boldsymbol{x}}\|_2 \\
&\leq \|\boldsymbol{H}_{\mathcal{T}-\tilde{\mathcal{T}}} \boldsymbol{x}_{\mathcal{T}-\tilde{\mathcal{T}}}\|_2 \\
&\leq \frac{1 + \delta_{3k}}{1 - \delta_{3k}} \frac{\sqrt{10\delta_{2k}}}{1 + \delta_{2k}} \|\boldsymbol{x}_{\mathcal{T}-\hat{\mathcal{T}}}\|_2 \\
&\leq (1 + \delta_{3k}) \frac{\sqrt{10\delta_{2k}}}{1 - \delta_{3k}} \|\boldsymbol{x}_{\mathcal{T}-\hat{\mathcal{T}}}\|_2
\end{aligned}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{H}_{\tilde{\mathcal{T}}}^{\dagger} \boldsymbol{y}$. In the third line of the above formula we have made use of of [26,

Th.3,Th.4]. In the last line Lemma 2 was applied.

By applying Lemma 4:

$$\begin{aligned}\|\tilde{\boldsymbol{y}}_r\|_2 &\geq \frac{1-2\delta_{3k}}{1-\delta_{3k}}\|\boldsymbol{y}\|_2 \\ &\geq \frac{1-2\delta_{3k}}{1-\delta_{3k}}\|\boldsymbol{H}_{\mathcal{T}-\hat{\mathcal{T}}}\boldsymbol{x}_{\mathcal{T}-\hat{\mathcal{T}}}\|_2 \\ &\geq (1-2\delta_{3k})\|\boldsymbol{x}_{\mathcal{T}-\hat{\mathcal{T}}}\|_2\end{aligned}$$

By combining the last two inequalities into (5.25) one arrives at at the following requirement:

$$\frac{1+\delta_{3k}}{1-2\delta_{3k}} \cdot \frac{\sqrt{10\delta_{3k}}}{1-\delta_{3k}} < 1.$$

After some basic computations we conclude that $\delta_{3k} < 0.205$. □

Theorem 14 concludes the theoretical analysis regarding the improvement of the inference algorithm by providing the sufficient condition under which the modified version of FMLM recovers all k-sparse signals exactly for a certain criterion. This condition is equivalent to the mutual coherence restriction for the OMP [91]. It also provides the sufficient condition for exact recovery for the SP-like variant of the algorithm.

5.4.3 Inference Algorithm Performance

To verify the preceding statements on the performance of the algorithms, a simple experiment is conducted. The algorithms under comparison are the FMLM algorithm as originally presented in [90], the variants based on the scaled quantities; FMLM-x_i, FMLM-α_i, FMLM-δl_i, FMLM-θ_i, the BSP and sparse recovery via linear programming. The OMP algorithm is also run for comparison with the variants. The results are acquired with the *cvx* software package [46] and are referred to as BP (Basis Pursuit). Noise variance is assumed to be $\sigma^2 = 0$. The experiment is as follows,

1. Generate $\boldsymbol{H} \in \mathbb{R}^{128\times 256}$ with i.i.d entries from $\mathcal{N}(0, \frac{1}{n})$.
2. Generate $\mathcal{T}$ uniformly at random so that $|\mathcal{T}| = K$.
3. Generate $\boldsymbol{x}_{\mathcal{T}}$ with i.i.d entries from $\mathcal{N}(0,1)$. Set $\boldsymbol{x}_{\mathcal{T}^c} = 0$, where $\mathcal{T}^c = \{[1,256] - \mathcal{T}\}$.
4. Compute $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ and then apply a reconstruction algorithm. Compare estimate $\hat{\boldsymbol{x}}$ to $\boldsymbol{x}$.
5. Repeat experiment 100 times for the same value of K.

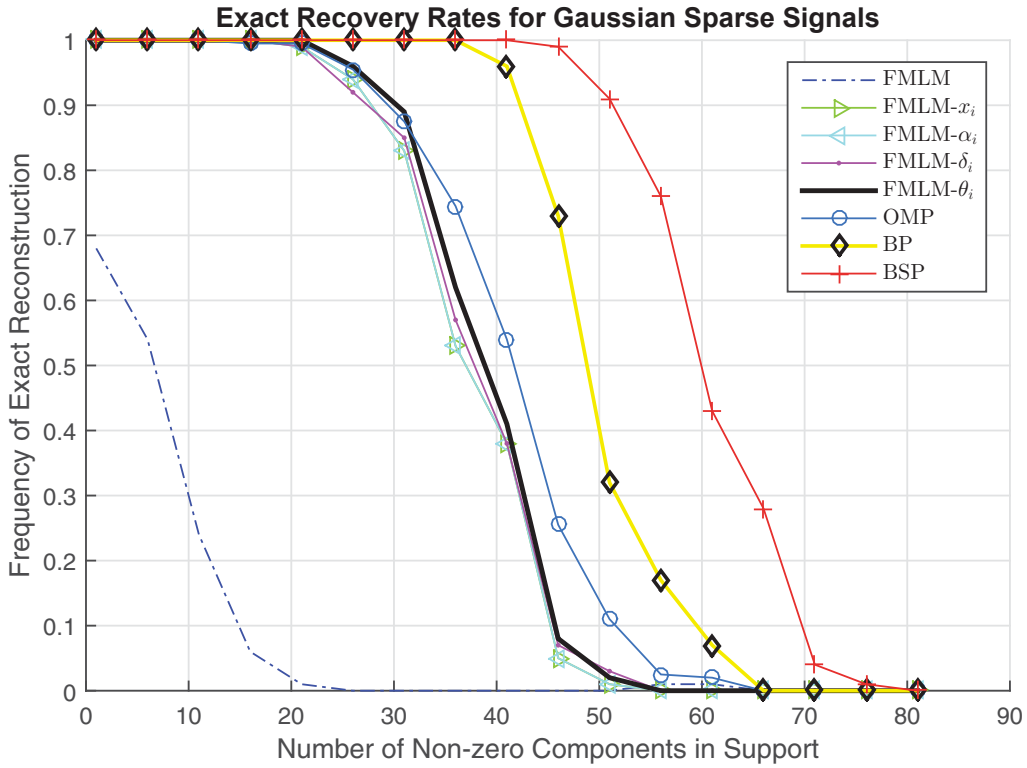


Figure 5.3: *Exact reconstruction rates for* $n = 128$, $m = 256$.

This experiment is run for different values of K. The results from this procedure are depicted in Figure 5.3. The first critical observation is that the original FMLM performs poorly when $\sigma^2 = 0$ due to the improperly scaled cost function. The three scaled variants of FMLM based on the criteria mentioned in Theorem 13 perform in the same manner. There is an increase in the performance for FMLM-θ_i, a consequence of altering the selection criterion to $\theta_i = \bar{q}_i$. Even though changing the criterion gives theoretically better performance as Theorem 14 suggests, empirically this gain is not great. Comparing with the OMP; one can see that indeed the OMP bears very close performance with the FMLM-θ_i variant as it was suggested by Theorem 13. By re-designing the inference algorithm based on ideas from the SP it is possible to achieve far better performance, as the curve for the BSP algorithm shows. The results for BP are in agreement with a similar experiment performed in [26] which compares the SP greedy algorithm and the BP. It is noted that in the case where the sparse components take their values from $\{-1, +1\}$ it was shown in [26] that BP does indeed outperform the BSP. An exhaustive study of the empirical recovery performance of the aforementioned algorithms and their comparison with BP is not in the scope of this work.

5.5 Conclusion

In this chapter the natural consequence of a model was presented; the algorithms to perform *inference* on that model. This is not an easy task to perform efficiently especially when large datasets are at hand. An easy way out is a greedy rendition instead of direct optimisation. This has been widely accepted by the community and has managed to provide excellent results in many scenarios of practical importance.

A large amount of work has also been conducted towards understanding the relationships between Sparse Bayesian Learning and ℓ_0 norm minimisation. The basic theorems have been presented in order to point out the significance of this algorithm for compressed sensing in general.

A refined theoretical analysis of the FMLM greedy inference algorithm was presented and is one of the major contributions of this thesis. In short, the *direct* connection of FMLM and greedy pursuits has been discovered. Moreover it was revealed that improvements are indeed possible and a set of algorithms has been provided that are less-greedy in performing inference in sparse Bayesian models.

Chapter 6

Dynamic Sparse Signal Recovery

In previous chapters, the signal acquisition process was modelled as a linear system of equations, $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$ with additive noise which is assumed to be in many cases white Gaussian. The previous chapter demonstrated the case where a *sparse* $\boldsymbol{x} \in \mathbb{R}^m$ is recovered from the noisy measurement vector $\boldsymbol{y} \in \mathbb{R}^n$ in the particularly interesting case where $n \ll m$. The signals under consideration were assumed to be static, i.e., the realisation of an experiment for only one time instant.

Let us put the problem of sparse signal recovery aside for a moment and consider the following simple example. Consider a volt meter providing a noisy measurement y for a voltage x plus an amount of noise. Given only one measurement y then the best estimate is actually $x = y$! Given several measurements of x all taken instantaneously, i.e., one has many volt meters, then a better estimate for x would be the sample mean over all measurements since the noise terms will tend to cancel out. Assume that this scenario changes so that the quantity of interest changes over time to x_t and temporal measurements y_t are taken. Unfortunately the simple time average would not suffice since one would get only a single estimate at time instant t. Furthermore this estimate would not accurately describe the time-varying signal x_t because of possible extreme values or periods of smooth activity. An easy way out would be the introduction of a time *window* so that the average would be taken only for a number of past consecutive samples. The problem is then taken to determine the length of such a window to capture the variations of the signal while suppressing noise at the same time.

From the example given in the previous paragraph it is evident that a unified approach is needed in order to reconstruct a signal which varies with time, i.e., dynamic signals. The best way to proceed is to consider a more sophisticated model for the measurement process which addresses this problem in a systematic and uncluttered way. Boasting extremely simple implementations and countless applications, the Kalman filter can be thought of as the lightweight champion for his task. The dynamic system model offers explicit estimation of the system's dynamics with low

computational complexity. The filter manages to provide good results even if there is a model mismatch between the actual and the assumed model.

Another major contribution of this thesis is presented in this chapter where the case of a time-varying system produces sparse signals is considered. The great disability that one faces is that traditional methods fail to recover sparse estimates for the signal given its temporal measurements. A novel idea is presented which manages to combine the virtues of the Kalman filter and Sparse Bayesian models.

6.1 The Kalman Filter

When a sample vector $\boldsymbol{y}$ is available then the estimate $\hat{\boldsymbol{x}}$ given by a *minimum mean squared error* (MMSE) estimator is optimal in the squared error sense. Such an optimal estimator becomes linear when the samples and the model parameters are assumed to follow the Gaussian distribution and the estimator is then given by the maximising the posterior of $\boldsymbol{x}$ given $\boldsymbol{y}$. In the case where no prior knowledge is available for $\boldsymbol{x}$ then the estimate is equivalent to the maximum likelihood estimate, i.e., MAP estimation with a uniform prior. This coincides with the solution given by the Wiener filter.

In the Kalman filter model is assumed for the time-varying $\boldsymbol{x}_t$. The Kalman filter estimates in an on-line fashion the *state* $\boldsymbol{x}$ of a time varying *system* via its noisy measurements $\boldsymbol{y}$.

6.1.1 The State-Space Model

The model for a system to accommodate the dynamic nature of signals is assumed to be a *discrete-time* state-space model. The model assumes two types of conditional distributions,

$$\begin{aligned} \boldsymbol{x}_t &\sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \\ \boldsymbol{y}_t &\sim p(\boldsymbol{y}_t|\boldsymbol{x}_t), \end{aligned}$$

where the time index has been introduced into the mathematical notation. The first distribution describes the time evolution of the system's state while the second one models the measurement process given the state. The model comes with two very important assumptions that might seem restrictive at first but in reality allow for a rich class of systems to be modelled. In Figure 6.1 the graphical representation of the model is shown in the form of a Bayesian network.

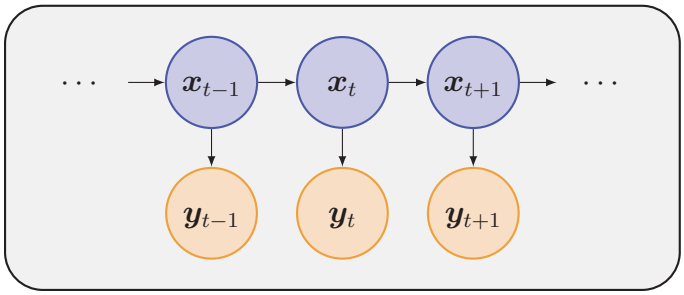


Figure 6.1: *The State-Space Model.*

The Markov property of the States

The Markov property dictates that a state at time t is only dependent on the previous state. In probabilistic terms this is expressed as

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}).$$

This is equivalent to the statement that whatever has happened in the past before $t-1$ will not directly affect the present state.

The Conditional Independence of the Measurements

The measurement at time instant t is conditionally independent of the previous states and measurements when conditioned on the state at the same time instant t. Again this is written as

$$p(\boldsymbol{y}_t|\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}) = p(\boldsymbol{y}_t|\boldsymbol{x}_t).$$

6.1.2 The Filtering Equations

The goal of the Kalman filter is to calculate the following posterior distribution

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$$

at each time instant t given the measurements up to and including t. The calculation of this marginal distribution can be seen as a two-step procedure.

The Prediction Step

In the prediction step one needs to find the distribution of $\boldsymbol{x}_t$ given $\boldsymbol{y}_{1:t-1}$. This can be accomplished as follows. At first the joint distribution of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{1:t-1}$ given

$\boldsymbol{y}_{1:t-1}$ is written down,

$$\begin{aligned} p(\boldsymbol{x}_t, \boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1}) &= p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) \\ &= p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}). \end{aligned}$$

where the Markov property has been used. The desired distribution is then formulated by *marginalising* over $\boldsymbol{x}_{t-1}$,

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}.$$

The *prediction step* projects forward in time the distribution of the previous state by using the dynamic state model. This way the assumed uncertainties of the state are elegantly taken into consideration. This uncertainty in the system's states is called *process noise*.

The Update Step

The *update step* basically is direct application of the Bayes' rule to find the distribution of $\boldsymbol{x}_t$ given the most recent measurement $\boldsymbol{y}_t$. The posterior distribution is written directly as,

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t}.$$

6.1.3 The Gaussian Assumption - Kalman Filter Equations

The Kalman filter assumes that the model is linear and described by the following set of equations,

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{F}_{t-1}\boldsymbol{x}_{t-1} + \boldsymbol{u}_{t-1} \\ \boldsymbol{y}_t &= \boldsymbol{H}_t\boldsymbol{x}_t + \boldsymbol{n}_t, \end{aligned}$$

where matrix $\boldsymbol{F} \in \mathbb{R}^{m\times m}$ is usually called the *state transition matrix* and it is used to describe the linear transform from one state to the next. The assumed distributions are Gaussian both for the measurement and the state model so in probabilistic terms the above equations are written as

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &= \mathcal{N}(\boldsymbol{F}_{t-1}\boldsymbol{x}_{t-1}, \boldsymbol{U}_{t-1}) \\ p(\boldsymbol{y}_t|\boldsymbol{x}_t) &= \mathcal{N}(\boldsymbol{H}_t\boldsymbol{x}_t, \boldsymbol{N}_t), \end{aligned}$$

where $\boldsymbol{U}_t$ and $\boldsymbol{N}_t$ are the covariance matrices of the state and the noise process respectively. The matrix $\boldsymbol{U}_t$ is assumed to be *known* or somehow estimated.

Since the assumed distributions are Gaussian the resulting conditional and marginal distributions mentioned above remain Gaussian. By using the direct result for the Gaussian distribution in Equation (A.3) of the appendix, the *prediction step* becomes

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) &= \mathcal{N}(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}) \\ \boldsymbol{\mu}_{t|t-1} &= \boldsymbol{F}_{t-1}\boldsymbol{\mu}_{t-1} \\ \boldsymbol{\Sigma}_{t|t-1} &= \boldsymbol{F}_{t-1}\boldsymbol{\Sigma}_{t-1}\boldsymbol{F}_{t-1}^T + \boldsymbol{U}_{t-1}, \end{aligned}$$

while the result in (A.4) gives the *update step* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ with

$$\begin{aligned} \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} - \boldsymbol{K}_t\left(\boldsymbol{y}_t - \boldsymbol{H}_t\boldsymbol{\mu}_{t|t-1}\right) \\ \boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_{t|t-1} + \boldsymbol{K}_t\boldsymbol{H}_t\boldsymbol{\Sigma}_{t|t-1}. \end{aligned}$$

Matrix $\boldsymbol{K}_t$ comes by the name of the *Kalman Gain* and is as follows,

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t\left(\boldsymbol{N}_t + \boldsymbol{H}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^T\right)^{-1}.$$

6.2 Dynamic Sparse Signals

In previous chapters it has been shown how the redundancy in the signal can be exploited in order to process the signal efficiently and moreover how to sample and reconstruct a sparse signal. In terms of this chapter; we have dealt with the *stationary* sort of sparse signals, i.e., a block of samples is taken and then a sparse estimator is sought. Focus is now turned on the case when such blocks are acquired one after the other and a sparse estimate of a signal is required in a temporal fashion.

Since modelling a dynamic system can be done efficiently and practically the natural consequence is to employ a dynamic system model to reconstruct *dynamic sparse signals*. A very convenient way to visualise this problem is in the form of a video signal where each separate frame is sparse in some domain (i.e. the wavelet domain). While reconstructing each frame separately would still exploit the spatial redundancy within each frame, the temporal correlation will remain unexploited. For a *dynamic sparse signal* $\boldsymbol{x}_t$ measured with

$$\boldsymbol{y}_t = \boldsymbol{H}_t\boldsymbol{x}_t,$$

the support set $\mathcal{T}_t$ is expected to be highly correlated with the support set at the

previous time instant $\mathcal{T}_{t-1}$, i.e.,

$$\mathcal{D}_t = \mathcal{T}_t - \mathcal{T}_{t-1}$$
$$|\mathcal{D}_t| \ll |\mathcal{T}_t \cup \mathcal{T}_{t-1}|$$

where the minus sign between sets denotes the set difference operation. The main assumption here is that the difference between the support sets between two consecutive sampling instants $\mathcal{D}_t$ is much smaller than either of the support sets. This is explained by the second equation. In order to take full advantage of the regularities in the dynamic signal, a linear estimator is required in order to track these small changes in the support. Unfortunately, direct application of the Kalman filter is inappropriate.

6.2.1 Incompatibilities and Limitations

- In Section 6.1 there was no assumption of sparsity and the prior distributions on the signals were assumed to be multivariate Gaussian. The solutions given by the estimator were allowed to take any form without any preference towards sparsity. In Chapter 2 it was demonstrated how sparse solutions can be promoted by imposing sparsity-promoting prior distributions on the signal to be recovered. The ML approaches effectively lead to regularisation based algorithms while MAP solutions provide a Bayesian approach to the sparse recovery problem.

- The Compressed Sensing literature mainly deals with the static case. It assumes batch computations on the complete dataset and based on a set of assumptions it provides algorithms along with performance guarantees for exact reconstruction. In general the sparse recovery machinery does not take into account the temporal correlation between datasets in order to improve sparse reconstruction as far as performance and computational requirements are concerned.

- When considering the tracking capabilities of the Kalman filter and the reasons why this technique is widely accepted as one of the most important in the field of filtering it is soon realised that it comes down to two factors. First, simplicity along with ease of implementation and second, optimality with respect to the MSE under mild conditions with acceptable performance when there is model mismatch. Keeping this in mind we recognise that the algorithms mentioned in Chapter 2 sometimes require some sort of sparsity regularising parameter or threshold in order to recover a sparse signal. This comes to contrast with the

two guidelines for dynamic sparse signal reconstruction since in a real-world dynamic scenario this luxury is highly likely to be unavailable.

- In Chapter 2 it was demonstrated that sparse signal sampling is possible to implement with efficient algorithms and acceptable performance guarantees in the case of noisy or even compressible sparse signals. This gives further motivation to pursue dynamic sparse signal tracking.

6.2.2 Related Approaches

Seminal work in this area is attributed to Vaswani and her work in [97]. A solution to the dynamic sparse signal recovery problem is proposed by what is an external modification of the Kalman filter. Sparse recovery is performed on the innovations signal whenever the prediction error rises above a certain threshold. Deletions from the estimated support set are also based on an additional threshold value that is suitably determined. The considered model is as follows:

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{H}_t \boldsymbol{x}_t + \boldsymbol{n}_t \\ \boldsymbol{x}_t &= \boldsymbol{x}_{t-1} + \boldsymbol{w}_t \\ (\boldsymbol{q}_t)_{T_t \cap T_{t-1}} &= \sigma^2_{sys} \\ (\boldsymbol{q}_t)_{T_t - T_{t-1}} &= \sigma^2_{init} \\ (\boldsymbol{q}_t)_{T_t^c} &= 0 \end{aligned}$$

where the time-varying sparse signal $\boldsymbol{x}$ is assumed to be a random variable drawn from $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\boldsymbol{q}_t))$. Set $\mathcal{T}_t$ denotes the support set of $\boldsymbol{x}_t$ and $\mathcal{T}^c$ means the complement of set $\mathcal{T}$ over its domain. The generative signal model assumes that changes in the sparse signal happen at random and these can be additions or deletions to its support set. Components of $\boldsymbol{x}_t$ that are zero can attain a non-zero value at $t+1$ with variance σ^2_{init} while the remaining non-zero components evolve with variance σ^2_{sys}. Observations $\boldsymbol{y}_t$ are taken at each time instant t via the measurement matrix $\boldsymbol{H}_t$ plus white Gaussian noise with variance σ^2_{obs}.

The algorithm runs a reduced order Kalman filter iteration based on the estimate of the support set $\mathcal{T}_{t-1}$ and then the filtering error is computed. If it is found to be above a certain threshold then a suitable sparse recovery algorithm is run to compute the support set of the error signal. The Kalman filter update step is run again on the merged support sets of the filtering error and $\mathcal{T}_{t-1}$. After this step yet another threshold is applied to delete all the small components indicating deletions in the support. If there have been any deletions the Kalman update is run *again*. The authors in [99] propose to replace the Kalman update in this algorithm with a Least Squares step. Also they provide the conditions under which the algorithms

converge to the case when the support is exactly known. Note that in these series of algorithms only changes in the support are considered and not in the actual magnitudes of the components in $\boldsymbol{x}_t$.

The authors in [98, 100] propose two similar ideas; to apply compressive sensing on the residual signal of the Kalman filter or the Least Squares residual signal. This time the authors take into consideration the tracked magnitudes as well as the support. An application for Magnetic Resonance Imaging (MRI) is documented in [77].

Vaswani and her team make yet another excellent contribution in [102] by recasting the dynamic sparse signal recovery problem as a compressed sensing problem with partially known support. Basically the following problem is formulated,

$$\min_{\boldsymbol{b}} \|\boldsymbol{b}_{\mathcal{T}^c}\|_1 \text{ so that } \boldsymbol{y} = \boldsymbol{H}\boldsymbol{b}$$

in which the best possible sparse representation or the $\boldsymbol{y}$ is desired but on a support set *complementary* to $\mathcal{T}$ which is considered to be the *prior knowledge*. The exact recovery conditions are derived and under certain assumptions it is shown that they are less restrictive than those for ℓ_1 minimisation for $\mathcal{T} = \emptyset$. The straightforward application of this to the dynamic case is also proposed with the addition of a threshold to detect changes. Further theoretical analysis on the stability of this method are given in [101] and an application of this on MRI in [59].

Finally in [60, 62, 61] introduce another extension of the aforementioned methods which basically introduces noise in the dynamic recovery procedure. It is also considered that some prior *knowledge* is available, i.e., that parts of the support set are known and that the magnitudes of the corresponding components are also given. It is also assumed that this *knowledge* can be partially erroneous. Analysis undertakes the following familiar problem,

$$\min_{\boldsymbol{b}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}\|_2 + \lambda\|\boldsymbol{b}_{\mathcal{T}^c}\|_1$$

which the authors term as *Regularized Modified Basis Pursuit De-noising*.

Another example of early work in dynamic sparse signal recovery was by Angelosante *et al.* and can be found in [2]. The authors propose two algorithms based on the LASSO. The first algorithm considers the case where the support remains stationary but the magnitudes are ever-changing. The second algorithm is completely dynamic and can recover both signals with time-varying support and magnitudes. The algorithm for the stationary case is termed *Group-Fused Lasso* minimises the

following cost function,

$$\min_{\boldsymbol{x}_1 \cdots \boldsymbol{x}_T} \sum_{t=1}^{T} \|\boldsymbol{y}_t - \boldsymbol{H}_t \boldsymbol{x}_t\|_2^2 \text{ so that}$$

$$\sum_{k=1}^{K} \sqrt{\sum_{t=1}^{T} |(\boldsymbol{x}_t)_k|^2} \leq s_1 \text{ and } \sum_{k=1}^{K} \sqrt{\sum_{t=2}^{T} |(\boldsymbol{x})_k - (\boldsymbol{x}_{t-1})_k|^2} \leq s_2.$$

Basically a number of sparse signals $[\boldsymbol{x}_1 \cdots \boldsymbol{x}_T]$ is recovered jointly under certain constraints. The first constraint encourages grouping of the components that have the same index in the support. A group is described as the non-zero components in $[\boldsymbol{x}_1 \cdots \boldsymbol{x}_T]$ that have the same index. The second constraint promotes smoothness ,i.e., that the components within the same group will have similar amplitudes. For the time-varying case the authors alter the cost function given above by adopting the ℓ_1 norm instead of the quadratic.

Some novel ideas that incorporate Bayesian modelling of time-varying signals are introduced by Ziniel *et al.* in [112]. A probabilistic model for the support is introduced based on a Bernoulli prior while the magnitudes evolve based on a Gauss-Markov process resembling a random walk model. The signal model is described by the equations below

$$(\boldsymbol{x}_t)_i = (\boldsymbol{s}_t)_i \odot (\boldsymbol{\theta}_t)_i \forall i \in [1, m]$$
$$(\boldsymbol{\theta}_t)_i = (1-a)(\boldsymbol{\theta}_{t-1})_i + a(\boldsymbol{w}_t)_i.$$

Symbol $\odot$ means element-wise multiplication. Random variables $(\boldsymbol{s}_t)_i$ are assumed to be drawn independently from the Bernoulli distribution with small probability of appearance so as to promote a sparse $\boldsymbol{x}_t$. Random variables $(\boldsymbol{w}_t)_i$ are i.i.d Gaussian with some variance and together with a deterministic $a \in [0, 1]$ they control the value of the non-zero components. Two transition probabilities are also considered for the case of components being added or deleted from the support. Signal amplitudes and support are estimated separately via a *Belief Propagation* algorithm. More details on this algorithm will be presented at a later chapter and the interested reader is redirected to the original paper.

Carmi *et al.* attempt to estimate a dynamic sparse signal with a modified Kalman filter *but by avoiding* the use of an internal compressed sensing algorithm like in the work of Vaswani and others. Their work can be found in [20]. This inspired idea suggest the notion of *pseudo-measurements*. Very simply the authors *embed* an ℓ_1 optimisation in the Kalman filter by augmenting the measurements by a fictitious

one

$$\begin{aligned} \mathbf{0} &= \|\boldsymbol{x}_t\|_1 - \epsilon \\ &= \bar{\boldsymbol{h}}_t \boldsymbol{x}_t - \epsilon \end{aligned}$$

where $\bar{\boldsymbol{h}}_t = \text{sign}(\boldsymbol{x_t})$ is the element-wise sign function. An additional update step is then run for this pseudo-measurement updating thus the statistics of $\boldsymbol{x}_t$.

The approach in [19] also employs a technique which requires several threshold values to be set which affect the quality of the solution.

In [22], a technique is presented that uses prior sparsity knowledge into the tracking process but also requires a number of parameters to be pre-set. The techniques revolve around the idea of casting the dynamic sparse signal recovery problem as a classic compressed sensing problem for one time instant. There is a discrimination between sparsity in the states and sparsity in the innovations. This work is closely related to the later work by Vaswani *et al.*

The approach in [24] extends the LMS algorithm to promote sparse system identification. The authors augment the cost function of the standard LMS with several cases of sparsity penalties. These penalty function incorporate several parameters that need defining priorly to the application.

Work in [66] proposes to alter the CoSaMP greedy pursuit algorithm [72] from block mode of operation into a sequential mode on-line algorithm. Because of this, it assumes *a-priori* known sparsity levels. Moreover a forgetting factor is employed in order to deal with time-varying sparse signals. The algorithm does not attempt to perform any further tracking of the statistics and no dynamic data model is assumed.

The framework developed in [42] deals with modelling highly complex dynamical linear systems something which falls well outside our cases of interest. The authors employ a multi-layered hierarchical probabilistic model able to capture discrete and continuous events. It is uncertain yet whether simplifications to this model can be used for compressed sensing. The complexity of the model makes analysis obscure.

To conclude; there has been some work which aimed in transforming LMS-type of algorithms but *it is not related* to dynamic sparse signals. Work in [50], [87] presents an ℓ_0 treatment of LMS type of algorithms along with performance guarantees to solve the sparse reconstruction problem in an adaptive filtering framework. However the dynamic sparse signal case is not considered at all and usually this sort of work gets confused because of the use of LMS-like algorithms.

6.3 The Hierarchical Bayesian Kalman Filter

The novel approach which was introduced in [55] is one of the contributions of this thesis. The dynamic sparse signal model which was proposed by the authors will be described and analysed. The *Hierarchical Bayesian Kalman Filter* (HBK) runs just like a traditional Kalman filter but with a *sparse DNA structure*. At the heart of this new technique is the efficient Type-II ML inference algorithm which was presented earlier in this text. To take things one step further, the authors in [56] have merged their improved inference algorithms into the HBK filter resulting in a superior sparse signal tracking algorithm.

6.3.1 A Hierarchical Model for Dynamic Sparse Signals

Let us start off this discussion by defining the grounds on which work will take place. It has already been pointed out that the structure of the Kalman filter is desirable and should be kept, in that it exhibits *the Markov property of the states* and *conditional independence of the measurements*. For this reason the dynamical system equations are kept the same with a slight alternation to fit our case of interest. That is; to accommodate a measurements model for sparse signals. In the proposed approach the following set of equations describe the dynamic system under consideration,

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{x}_{t-1} + \boldsymbol{u}_{t-1} \\ \boldsymbol{y}_t &= \boldsymbol{H}_t \boldsymbol{x}_t + \boldsymbol{n}_t, \end{aligned} \tag{6.1}$$

The measurement process is considered to be Gaussian with known covariance matrix $\boldsymbol{n}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. It is assumed that signal $\boldsymbol{x}_t \in \mathbb{R}^m$ is sparse in some domain and this sparsity domain is considered to remain unchanged at all time instants. This allows to set the state transition matrix $\boldsymbol{F}_t$ equal to the unitary matrix $\boldsymbol{I}$.

In Figure 6.2 the graphical representation of the proposed dynamic model is given. The main attribute of this work is the obvious extension of the classic state-space dynamic model; by the inclusion of another level of prior distributions. This is of course the same sort of hierarchical model that was met in Sparse Bayesian Learning. A sparsity promoting prior distribution is used to model the states. After performing inference the resulting $\boldsymbol{x}_t$ will be sparse. More details on this will follow.

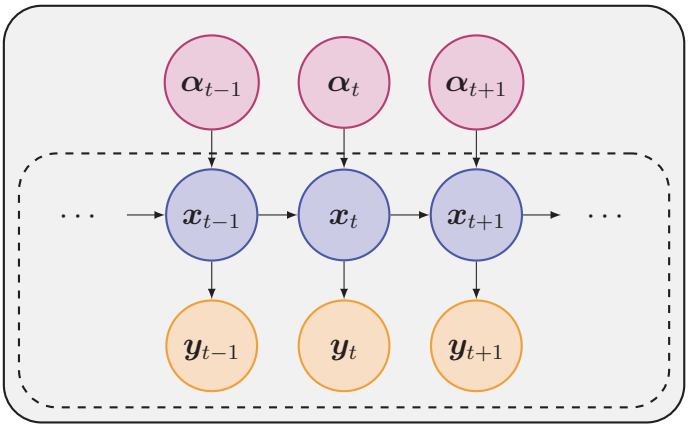


Figure 6.2: *The Hierarchical Bayesian Kalman filter Bayesian network. The state random variables are further described by a hierarchical prior structure that is met in sparse Bayesian learning models.*

6.3.2 The Revised Prediction and Update Steps

The proposed Kalman filter model bears many similarities when it comes to temporal passing of correlations but the actual expressions for the update and prediction steps have to be altered. It all comes down to marginalisation at some point and the expanded prior structure dictates to recompute those marginals. It was stated above that the noise variance is considered to be given. It is already known from Sparse Bayesian Learning that it can be estimated from the data. This is not our main concern and without any sacrifices it will not occupy us any further.

In the Hierarchical Bayesian Kalman Filter it assumed that $\boldsymbol{u}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{A}_t^{-1})$ where $\boldsymbol{A}_t = \text{diag}(\boldsymbol{\alpha}_t) = \text{diag}([\alpha_1, \cdots, \alpha_n]_t)$ and hyper-parameters $\boldsymbol{\alpha}_t$ have to be learned by optimising the cost function. Driving $\alpha_{t,i} \to +\infty$ results in $p(u_{t,i}|\alpha_{t,i}) \to \delta(0)$ which means that it is *a posteriori* certain that $u_{t,i} = 0$. The Kalman filter two-step procedure is still performed with slight alternations so as to accommodate the revised system model.

In the *prediction step* the parameters of $p(\boldsymbol{x}_t|\boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$ are evaluated straightforwardly as follows

$$\begin{aligned} \boldsymbol{\mu}_{t|t-1} &= \boldsymbol{\mu}_{t-1}, \\ \boldsymbol{\Sigma}_{t|t-1} &= \boldsymbol{\Sigma}_{t-1} + \boldsymbol{A}_t^{-1}, \end{aligned} \tag{6.2}$$

For the *update step* the following quantities are defined,

$$\begin{aligned} \boldsymbol{y}_{t|t-1} &= \boldsymbol{H}_t \boldsymbol{\mu}_{t|t-1}, \\ \boldsymbol{y}_{e,t} &= \boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}. \end{aligned}$$

The parameters of $p(\boldsymbol{x}_t|\boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ can then be written down as,

$$\begin{aligned} \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \boldsymbol{K}_t \boldsymbol{y}_{e,t}, \\ \boldsymbol{\Sigma}_t &= (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t)\boldsymbol{\Sigma}_{t|t-1} \\ \boldsymbol{K}_t &= \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^T(\sigma^2\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}^T)^{-1} \end{aligned} \tag{6.3}$$

6.3.3 The HBK Inference Algorithm

From Equation (6.2) it is straightforward that,

$$\begin{aligned} \boldsymbol{y}_{e,t} &= \boldsymbol{y}_t - \boldsymbol{y}_{t|t-1} \\ &= \boldsymbol{H}_t \boldsymbol{u}_t + \boldsymbol{n}_t. \end{aligned}$$

It becomes clear that a sparse $\boldsymbol{u}_t$ has to be inferred from the available data. Effectively a sparse *prediction error* signal has to be recovered at each time instant which will also - very likely - produce a sparse $\boldsymbol{x}_t$.

In order to recover the needed statistical information from $\boldsymbol{y}_{e,t}$ there is an additional step of learning $\boldsymbol{\alpha}_t$. By incorporating the prediction step $p(\boldsymbol{x}_t|\boldsymbol{y}_{t-1})$ the following posterior needs to be minimised,

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}_t) &= \log p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t) = \log \mathcal{N}\left(\boldsymbol{H}_t\boldsymbol{\mu}_{t-1}, \boldsymbol{C}_u\right) \\ &= \log p(\boldsymbol{y}_{e,t}|\boldsymbol{\alpha}_t) = \log \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{C}_u\right) \\ &= \log|\boldsymbol{C}_u| + \boldsymbol{y}_{e,t}^T \boldsymbol{C}_u^{-1} \boldsymbol{y}_{e,t} \end{aligned} \tag{6.4}$$

where $\boldsymbol{C}_u = \sigma^2\boldsymbol{I} + \boldsymbol{H}_t\left(\boldsymbol{\Sigma}_{t-1} + \boldsymbol{A}_t^{-1}\right)\boldsymbol{H}_t^T$.

If we compare the cost function above with Equation (5.3) we notice that there is a difference on the covariance matrices involved. Let us focus on matrix $\boldsymbol{C}_u$. This can be written as,

$$\begin{aligned} \boldsymbol{C}_u &= \sigma^2\boldsymbol{I} + \boldsymbol{H}_t\left(\boldsymbol{\Sigma}_{t-1} + \boldsymbol{A}_t^{-1}\right)\boldsymbol{H}_t^T \\ &= \sigma^2\boldsymbol{B} + \boldsymbol{H}_t\boldsymbol{A}_t\boldsymbol{H}_t^T \end{aligned}$$

where $\sigma^2\boldsymbol{B} = \sigma^2\boldsymbol{I} + \boldsymbol{H}_t\boldsymbol{\Sigma}_{t-1}\boldsymbol{H}_t^T$. By applying the matrix inversion lemma [44],

$$\sigma^2\boldsymbol{C}_u^{-1} = \boldsymbol{B}^{-1} - \boldsymbol{B}^{-1}\boldsymbol{H}_t\left(\sigma^{-2}\boldsymbol{\Sigma}_u\right)^{-1}\boldsymbol{H}_t^T\boldsymbol{B}^{-1} \tag{6.5}$$

where $\sigma^{-2}\boldsymbol{\Sigma}_u = \left(\sigma^2\boldsymbol{A}_t + \boldsymbol{H}_t^T\boldsymbol{B}^{-1}\boldsymbol{H}_t\right)$ is the scaled covariance matrix of $\boldsymbol{u}_t$. This way the contribution from the previous time instant in computing matrix $\boldsymbol{C}_u$ has been quantified exactly. Bayesian inference can now be performed the same way as it was described in Chapter 5. Please note the use of *scaled* quantities as in Chapter

Algorithm 7 HB-Kalman Filter

Input: $\boldsymbol{y}_t$, $\boldsymbol{H}_t$, σ^2
Initialise:
- $\mathcal{T}_1 = \{\text{index } i \in [1, n] \text{ for maximum } |\boldsymbol{h}_{t,1}^T \boldsymbol{y}_1|\}$.
- Calculate $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$ with support set $\mathcal{T}_1$.

Iteration:
- Apply *Prediction Step* by calculating $\boldsymbol{y}_{e,t}$.
- Apply InferenceAlgorithm($\boldsymbol{y}_{e,t}$, $\boldsymbol{H}_t$, σ^2, $\boldsymbol{\Sigma}_{t-1}$) to recover $\boldsymbol{u}_t$, $\boldsymbol{\Sigma}_u$ with support set $\mathcal{I}_u$.
- Expand or contract $\mathcal{T}_t$ accordingly.
- Calculate $\boldsymbol{\Sigma}_{t|t-1}$.
- Apply *Update Step* and calculate $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$.

Output:
- At time instant t, output sparse signal $\boldsymbol{x}_t$ with mean value $\boldsymbol{\mu}_t$, covariance matrix $\boldsymbol{\Sigma}_t$ and support set $\mathcal{T}_t$.

5. Assuming scaled quantities also makes the situation easier when incorporating the *improved* inference algorithms that where also introduced.

In Algorithm 7 the steps of the HBK filter are shown. In the second step of the iteration part the inference algorithm to be employed is shown as a sub-routine, InferenceAlgorithm. This means that at this point optimisation of Equation (6.4) takes place. This sub-routine should take as input the prediction error signal, the noise variance, the sampling matrix and the covariance matrix from the previous time step $\boldsymbol{\Sigma}_{t-1}$. This is done so that matrix in Equation (6.5) can be computed.

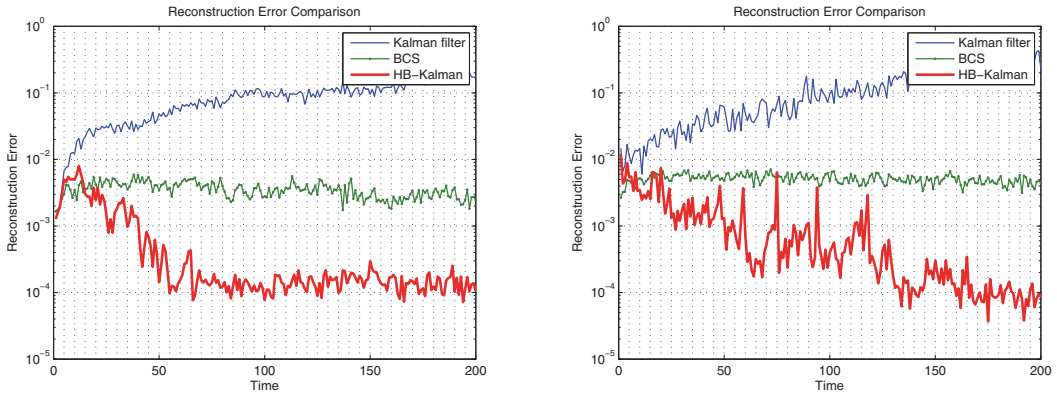
Algorithms 5 and 6 can be used as the InferenceAlgorithm subroutine in the description of Algorithm 7. Note that Algorithm 5 and Algorithm 6 are not affected by this modification except from the computation of the relevant quantities like s_i and q_i with the new matrix in Equation (6.5). The Kalman filter steps given above can be used to track the scaled covariance matrix of sparse signal $\boldsymbol{x}_t$.

The HBK filter steps are described in Algorithm 7. A sparse support for $\boldsymbol{y}_{e,t}$ is recovered and the support set of the dynamic signal is updated accordingly. The use of InferenceAlgorithm as a sub-routine in the algorithm description refers to an appropriate algorithm to perform inference and produce the necessary statistical information.

6.3.4 HBK Filter Advantages

- The whole mechanism remains agnostic about the sparsity level of the dynamic signal. The filter tracks the support set along with the magnitudes of the sparse components in a unified manner. This retains the original nature of the Kalman filter; to be simple to implement and efficient to compute.

- The HBK filter does not rely on any external modifications or controlling



(a) *Measurements* $n = 100$. **(b)** *Measurements* $n = 28$.

Figure 6.3: *Tracking performance comparison between the HB-Kalman and the classic Kalman filter for sparse input. The i.i.d case is provided for comparison and verification. (a) Number of noisy measurements is adequately high to ensure reconstruction. (b) Number of measurements is reduced to unsustainable levels that do not allow exact reconstruction. HB-Kalman successfully tracks the sparse signal by employing temporal information gained from tracking.*

parameters (except of course from the measurement noise variance which is assumed to be known or estimated). Sparsity is realised in a Bayesian framework where the internal inference algorithm requires no external parameters.

- The HBK filter carries the same sort of inference algorithms like the ones presented in Chapter 5. This means that automatically the HBK filter comes attached to a set of provable performance guarantees and sufficient criteria for recovery.

6.4 Test Cases

In this section several empirical test cases will be presented. These experiments have been published in [39, 54]. Three such experiments are discussed; one in which synthetic data are used as input to the algorithms to be tested, a real-world scenario in which incomplete Ozone data are treated to complete the missing samples and finally a rather ambitious case in which an attempt is made for audio signal reconstruction after the signal has been synthetically corrupted.

6.4.1 Synthetic Scenarios

The original Kalman filter and the proposed method are compared in a case of synthetically generated dynamic sparse signals. The case where the samples are assumed to be independent and identically distributed is also considered, i.e. apply the Bayesian Compressive Sensing (BCS) algorithm [84] independently at

each time instant. This example aims to underscore that temporal correlations can be exploited towards achieving better sparse recovery performance as opposed to when applying compressed sensing independently at each time instant.

Signal $\boldsymbol{x}_t \in \mathbb{R}^m$ is assumed to be sparse in the natural basis with support set $\mathcal{S}$ chosen uniformly at random from $[1, m]$ where $m = 256$. The magnitudes of the non-zero entries of $\boldsymbol{x}_t$ evolve according to Equation (6.1) with $\boldsymbol{U}_t = \sigma_u^2 \boldsymbol{I}$ with $\sigma_u^2 = 0.1$. The simulation time for this experiment was 200 time instants. Noisy measurements $\boldsymbol{y}_t$ with the entries of matrix $\boldsymbol{H}_t \in \mathbb{R}^{128\times 256}$ being drawn from $\mathcal{N}(0, \frac{1}{n})$ and to be re-sampled at each time instant. Measurement noise variance is set to $\sigma^2 = 0.01$ for the entire simulation time. At two randomly chosen time instants; $t = 50$ and $t = 150$, a change in the support of $\boldsymbol{x}_t$ is introduced. A non-zero component is added to the support of $\boldsymbol{x}_{50}$ and a non-zero component is removed from the support of $\boldsymbol{x}_{150}$. Apart from these two time instants the support of $\boldsymbol{x}_t$ remains unaltered. At $t = 1$ the support is initialised with $k = 30$ non-zero components.

In Figure 6.3a the reconstruction error is plotted against time for each of the three reconstruction methods. It is evident that the error levels are much lower for the HB-Kalman filter when compared to the conventional Kalman filter, a direct consequence of the assumed sparse model. By comparing to the repeated application of the BCS method it is demonstrated that by incorporating statistical information from previous estimates results in lower reconstruction error.

In a more difficult setting, the number of measurements is reduced but the sparse signal is of a sparsity level well above this number. More specifically, $n = 28 < 2\times 30$ is less than twice the number of active components which is 30. This is a particularly difficult case since the number of measurements is less than what is required for exact reconstruction of the sparse signal. The sparse signal $\boldsymbol{x}_0$ is taken to be known beforehand. This corresponds to the case where successful reconstruction with an adequate number of measurements has been achieved but for some external technical reason the number of measurements is forced to be reduced to what usually are unsustainable low levels. It is shown in Figure 6.3b that given statistical information from an earlier time instant the filter manages to maintain its performance even though it might take slightly longer to converge.

6.4.2 The Ozone Distribution Dataset

The proposed method is tested on a *real-life* scenario. We attempt to track the spatial distribution of the Ozone layer over the entire globe. The dataset on which the proposed method is tested is obtained from the Ozone Monitoring Instrument (OMI) on the NASA Aura spacecraft [71]. The dataset consists of daily measurements for a number of months. This dataset can conceptualised as a cube on which

the $x-y$ dimensions represent the pixels of an image while the z dimension represent the time (days). This dataset is of particular interest since the measurements for each day of the month are *incomplete* due to a fault of the monitoring system. This can be seen in Figure 6.4a by the blue vertical stripes.

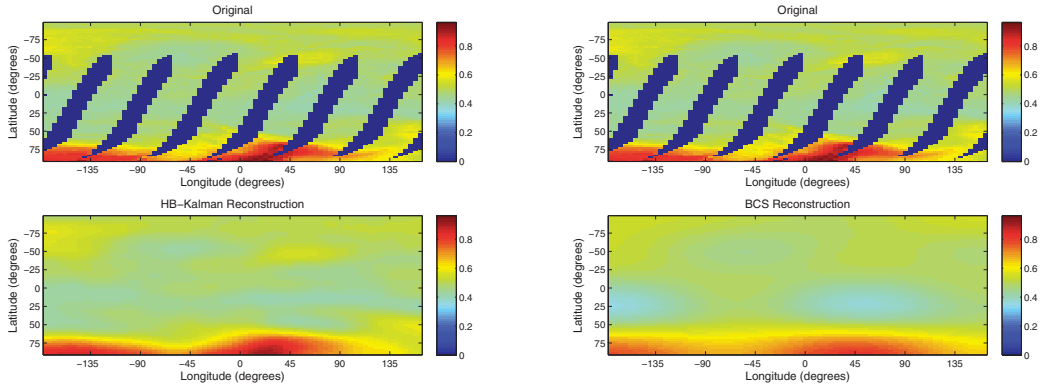
For the purpose of exposition it is considered that these heatmap-like images exhibit a sparse representation in the discrete cosine transform (DCT) domain. To be pedantic one should better say that these images are compressible. Possibly better results could have been achieved had a better representation domain was employed. The original images are cropped to form a square image and are under-sampled by a factor of 8 so that $n = 4275$. This is done so as to be able to perform the tests on a personal computer. Pausing for yet another moment, the value of sparsity is recognised when dealing with *datasets*. This however does not hinder performance and does not affect the generality of the results. The areas of the image that appear blue on the top of Figure 6.4 are the parts of the dataset that are missing due to the malfunction. The part of the dataset that was not damaged is assumed to take the role of the measurements $\boldsymbol{y}_t$ in the HB-Kalman filter model. The measured Ozone image for each day was transformed into a stacked vector by keeping only the undamaged data points (or pixels); the indices of which were also used to sample the corresponding rows of the DCT matrix. This sub-sampled DCT matrix plays the role of $\boldsymbol{H}_t$. Each time instant corresponds to one day of the month for a total of 28 days. Measurement noise variance is set to $\sigma^2 = 10^{-6}$.

By performing dynamic sparse signal recovery the support set of each image is recovered and subsequently a reconstructed image with the damaged pixels being filled-in. The accuracy of the reconstruction is then measured by computing the MSE between the pixels of the two images corresponding to the undamaged parts.

As can be seen from Figure 6.5, the standard Kalman filter fails to accurately track the dynamic sparse signal. By contrast, repeated application of the BCS method and the HB-Kalman filter, exhibit much lower error levels and accurately reconstruct the missing data. The HB-Kalman outperforms the BCS method as it incorporates statistical information from the previous day resulting in lower reconstruction error. By carefully comparing the results in Figure 6.4 it can be seen that the BCS reconstruction lacks some of the higher frequency components that are present in the HB-Kalman reconstruction and the original signal. This results in losing some of the details in the signal hence producing lower quality results.

6.4.3 Audio Signal Reconstruction

Here the rather ambitious problem of reconstructing a corrupted audio signal is considered; the recording of a classical piano piece in a real reverberant environment.



(a) *Reconstruction with HB-Kalman filter.* **(b)** *Reconstruction with BCS.*

Figure 6.4: *Reconstructed ozone distribution signal using the HB-Kalman filter and BCS. Atmospheric ozone distribution is measured in normalised Dobson units. Original data is shown on the top of each graph. For brevity only one frame from the complete reconstructed dataset is shown. The frame is the same for both cases.*

The recorded signal is highly non-stationary, broadband and contains overlapping notes (that might be harmonically related). To make things worse, the pedal on the piano is engaged throughout, causing significant time-frequency smearing. The piano recording is sampled at a frequency of $f_s = 44.1$ kHz and split into T non-overlapping frames of length $m = 1024$ samples.

The Fourier domain support of each frame is assumed to be *approximately sparse*, i.e., there is only a small number of dominant frequencies present, as can be seen from Figure 6.6. For each of these frames only a small number of samples are kept in order to artificially corrupt the signal. The indices of the samples that are kept are chosen uniformly at random from $[1, 1024]$ to form the index set $\mathcal{I}_t$ where t represents a time frame. Sampling matrix $\boldsymbol{H}_t = \mathcal{F}_{\mathcal{I}_t}^{-1}$ is then formed by choosing those rows from the Fourier matrix that correspond to $\mathcal{I}_t$.

At each time instant an estimate for the support $\boldsymbol{x}_t$ is recovered. Since the assumed basis is the Fourier basis the support is tracked in the both the real and imaginary domain. The measurement matrices now become $\Re\{\boldsymbol{H}_t\}$ and $\Im\{\boldsymbol{H}_t\}$. Note that these matrices belong in $\mathbb{R}^{n\times m/2}$ because of the symmetry of the Fourier transform for real-valued functions. The simulation time for this experiment is $T = 100$ time domain frames. At each time instant $m = 256$ samples are kept. Measurement noise variance is set to the sufficiently small value of $\sigma^2 = 0.1^5$ for the entire simulation time since the input signal was generated with no additional noise.

The resulting Root Mean Squared Error (RMSE) for the whole simulation time T is shown in Figure 6.8. As can be seen from the resulting graph, the error levels are much lower for the HB-Kalman filter when compared to the standard Kalman filter; this is a direct consequence of the assumed sparse model. By comparing to the repeated application of the BCS method, i.e assuming independent, identically

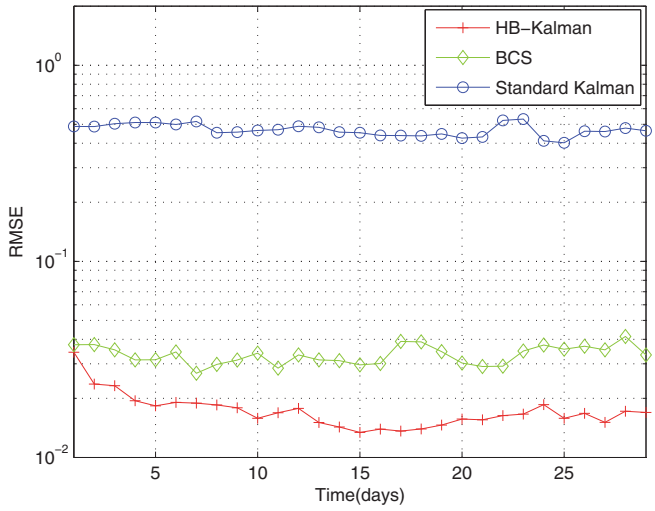


Figure 6.5: *Reconstruction error for the Ozone distribution dataset for the period of 28 days.*

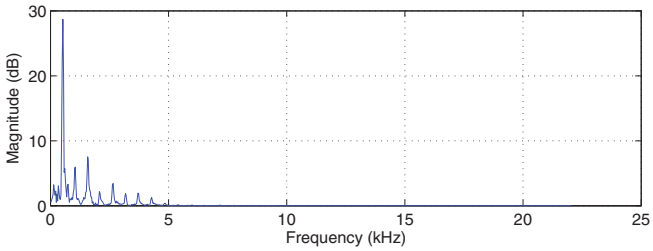


Figure 6.6: *Snapshot of the sparse frequency content in a single time-frame of piano data.*

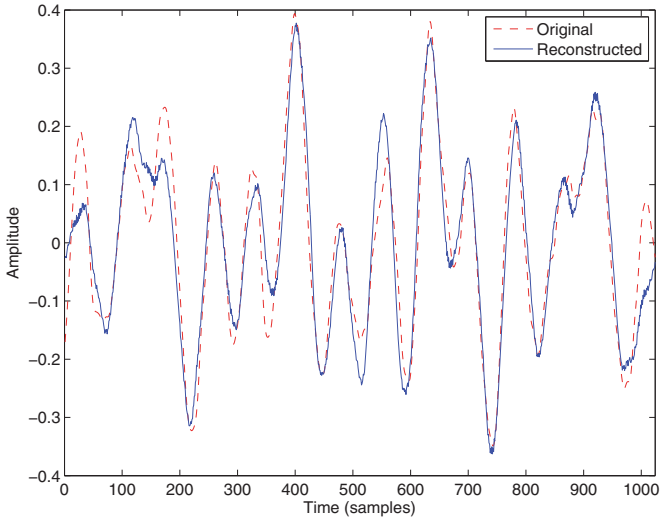


Figure 6.7: *Original time-domain representation (dotted line) of a frame of audio, along with the reconstructed data using the HB-Kalman.*

distributed data, we see that incorporating statistical information from previous estimates results in lower reconstruction error. In Figure 6.7 a comparison is made

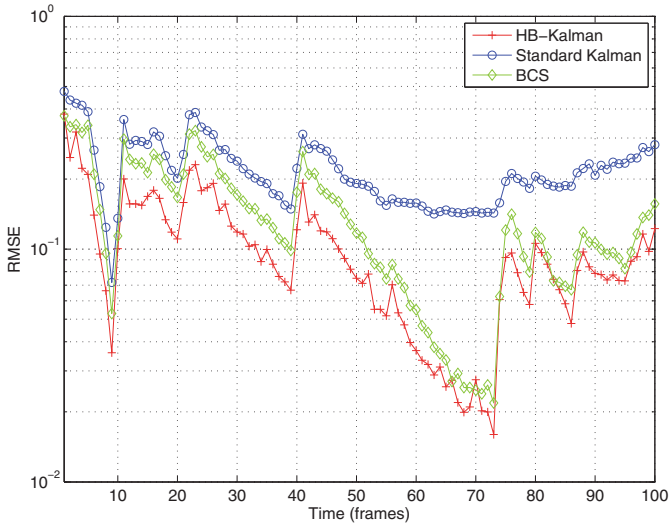


Figure 6.8: *Reconstruction error given over a time period of 100 frames. Each frame contains* $n = 1024$ *samples from which* $m = 256$ *are chosen at random.*

between the reconstruction of a frame and the original one. Even thought the two seem to be quite close, the perceived quality by the human ear is not that great. It also is important to emphasize that the parameters are identical for both the BCS algorithm and the HB-Kalman for a fair comparison; tuning certain parameters individually for each algorithm can lead to better reconstruction results depending on the specific application scenario.

The results are cannot be described as fascinating even though a basic point has been made. Working with audio signals is not at all straightforward particularly for this example. The dictionary that was chosen for sparse recovery was admittedly very naive and simple to capture all the transients and changes in the frequency domain of each frame. Nevertheless, positive results with such basic scenarios only show the potential of the proposed methods.

6.5 Conclusion

In this chapter the notion of *dynamic sparse signals* was introduced. As it happens in most cases, many classes of signals exhibit temporal correlations between their samples. It is therefore, only natural to ask whether it is possible to do any better for dynamic sparse signals, i.e., exploit this temporal redundancy to improve performance. The classic Compressed Sensing framework does not make provisions for the dynamic case and the well established sparse reconstruction algorithms perform reconstruction on a batch of samples which is considered to have been acquired instantaneously.

By employing any of the traditional adaptive estimators in an "out of the box"

fashion results in non-sparse solutions since the assumed probabilistic model fails to accurately model sparse systems. The employment of an appropriate probabilistic model in the Kalman filter system model allows us to surpass this problem and achieve full dynamic sparse signal tracking without any external modification or dangerous parameters that hinder performance. The resulting Hierarchical Bayesian Kalman Filter is a completely automatic sparse recovery framework which exploits temporal correlations between continuous samples. The filter has been shown to maintain the same flexibility as its non-sparse cousin.

The theoretical analysis of this framework highlights that it is possible to derive performance guarantees regarding the global optimality of the provided solutions. The proposed framework provides the flexibility to consider different probabilistic models with different prior distributions even with a completely redesigned Bayesian inference algorithm.

Chapter 7

Informative Sparse Bayesian Learning

In Chapters 4 and 5 a very versatile Bayesian model for sparse signals was analysed along with some very efficient inference algorithms. To summarise; every sparse component x_i is modelled as a separate Gaussian random variable with zero mean and variance α_i^{-1}, $\mathcal{N}(0, \alpha_i^{-1})$. A so-called *inference algorithm* is then employed to recover what the optimal value for the variance of each component is. After convergence one hopes that many of the variances will reach zero or near-zero values thus producing a sparse signal $\boldsymbol{x}$. It was demonstrated in earlier chapters that this model is indeed easily extended to cope with many different distributions and structures.

Let us undertake the case where not a simple point estimate for the value of α_i is sought but a *complete distribution* instead. Recalling from Chapter 4 each variance was attached to a hyper-prior distribution $p(\alpha_i) = \text{Gamma}(a, b)$ (the Gamma distribution) with all of the hyper-prior distributions sharing the same constant hyper-parameters a, b. In the so-called *truly Bayesian* approach the Bishop *et al.* in [10] consider that $p(\alpha_i) = \text{Gamma}(a_i, b_i)$ and adopt a special type of inference to compute the optimal values for the hyper-parameters of the distributions. The *Variational Sparse Bayesian Learning* will be explained below in more detail. The authors expected that the gains from this approach will be more pronounced in cases where the dataset size is limited despite the increased computational demands of the variational method.

From what was demonstrated in Chapter 5 in most of the cases it makes more sense to assume no prior knowledge about the sparse components x_i and in this discussion this translates to adopting an *uninformative prior* for the components' variance α_i. This is implemented by setting $a = b = 0$ in the traditional SBL or $a_i = b_i = 0$ in the variational SBL. In Section 5.1 it was explained that an uninformative prior allows for the posterior $p(\boldsymbol{\alpha}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})$ to be approximated in a certain way and basically assume that $p(\boldsymbol{\alpha}|\boldsymbol{y}) \approx \delta(\boldsymbol{\alpha}^*)$ since $p(\boldsymbol{\alpha})$ is taken to be

constant, i.e., uninformative and the most probable value $\boldsymbol{\alpha}^*$ is sought. The reader is redirected to Figure 4.5 for intuition. Based on the assumption for an uninformative prior, the authors in [86] have taken advantage of this special case to derive a fast variational inference algorithm. It was also proven that their fast variational algorithm is in fact equivalent to the fast Type-II maximum likelihood maximisation algorithm described in Chapter 5.

The main focus of this chapter is those cases for which an *informative prior* is used to model some preference over the distribution of each of the sparse components. Such information usually comes from empirical tests or expert systems. To incorporate such belief into the aforementioned hierarchical Bayesian model one has to finely-tune the distribution of each α_i and then run a suitable variational inference algorithm that will allow for an estimate $p(\alpha_i) = \text{Gamma}(\tilde{a}_i, \tilde{b}_i)$ to be found. It is explained that in the case where $a_i \neq 0$ and $b_i \neq 0$ the fast algorithm of [86] cannot be used. A theoretical analysis is conducted which explains this phenomenon for an *informative prior*. The analysis proceeds in establishing several modifications and rules that allow for a fast variational algorithm to exist for *informative prior distributions*.

7.1 Revisiting the Sparse Bayesian Model

Let us point out the main difference between the two approaches by using their corresponding graphical models. Figure 7.1 shows the graphical model for the variational approach to Sparse Bayesian Learning (VSBL), i.e., when the Gamma distribution parameters a_i, b_i are to be estimated instead of being fixed. Quoting the authors [10]; this results in the computation of the *distributions* for each α_i instead of point estimates. Comparison should be made against Figure 4.3 from Chapter 4. One quickly notices that inside the dashed rectangle (or *plate* in statistician's jargon) now lie the hyper-parameters as well indicating that now they have become part of the inference process. Recall that $p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2) = \mathcal{N}(\boldsymbol{H}\boldsymbol{x}, \sigma^2\boldsymbol{I})$.

Recalling from SBL, in order to compute the optimal values for α_i in such a hierarchical model, the *Evidence Procedure* is followed or *Type-II Maximum Likelihood*. This involves maximisation of the hyper-parameter posterior,

$$\log p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2).$$

In the case of an uninformative prior, i.e., distributions $p(\boldsymbol{\alpha}), p(\sigma^2)$ are essentially flat constant functions, one only maximises the likelihood $p(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma^2)$. For the considered cases of an *informative prior*, i.e., the prior distributions are no longer flat, this

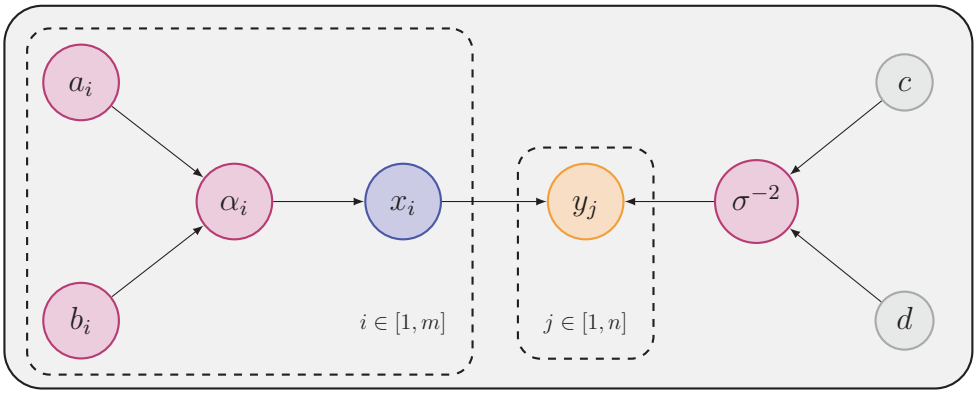


Figure 7.1: *Bayesian Network of the Variational Relevance Vector Machine.*

results in the following cost function,

$$\begin{aligned}\mathcal{L} &= \log p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{y}) = \log p(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma^2) + \log p(\boldsymbol{\alpha}) \\ &= \frac{1}{2}\left[\log\left|\sigma^2\boldsymbol{I} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T\right| + \boldsymbol{y}^T\left(\sigma^2\boldsymbol{I} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T\right)^{-1}\right] \\ &+ \sum_{i=1}^{m}\left(a_i \log b_i + (a_i - 1)\log\alpha_i - b_i\alpha_i - \log\Gamma(a_i)\right).\end{aligned} \tag{7.1}$$

To keep things uncluttered in the expression above, it was silently assumed that the noise variance is known and not a random variable.

One quickly realises that direct optimisation of the cost function for recovering the variances α_i is indeed tractable while optimisation with respect to hyperparameters a_i and b_i becomes more troublesome. In the section to follow the variational method is described which adopts a different and more holistic approach towards performing inference.

7.2 Variational Sparse Bayesian Learning

In this section the Variational Sparse Bayesian Learning of [10] is presented. Variational methods [51, 78] generally behave differently than the Type-II Maximum likelihood and act more like the Expectation-Maximisation algorithm [9] towards achieving approximate inference. The variational approach proceeds in defining a distribution Q which will be used to approximate the posterior,

$$p(\boldsymbol{x}, \boldsymbol{\alpha}|\boldsymbol{y}) \simeq Q(\boldsymbol{x}, \boldsymbol{\alpha})$$

It was shown in Chapter 5 and Equation (5.1) that this posterior is in fact intractable to compute. The *Variational Lower Bound* $\mathcal{L}(Q)$ is then maximised over all possible

distributions Q that follow a special form,

$$\mathcal{L}(Q) = \int Q(\boldsymbol{x}, \boldsymbol{\alpha}) \log \frac{p(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{y})}{Q(\boldsymbol{x}, \boldsymbol{\alpha})} d\boldsymbol{x} d\boldsymbol{\alpha}$$
$$Q(\boldsymbol{x}, \boldsymbol{\alpha}) = Q_{\boldsymbol{x}}(\boldsymbol{x}) Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}).$$

The special form of the approximating distribution Q is that it can be factorised while no other limitations are placed regarding the form of the factors. The lower bound is then optimised directly with respect to each of the factors. This might seem daunting at a first glance but as it turns out the resulting expressions are simple.

7.2.1 Variational Inference

Before proceeding it is re-iterated that one separate hyper-parameter for each α_i is contemplated to control the variance of each component x_i:

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{m} \text{Gamma}(a_i, b_i)$$
$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{i=1}^{m} \mathcal{N}\left(0, \alpha_i^{-1}\right) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{A}^{-1}\right)$$
$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n} p(y_i|\boldsymbol{x}, \sigma^2) = \mathcal{N}(\boldsymbol{H}\boldsymbol{x}, \sigma^2 \boldsymbol{I})$$

where matrix $\boldsymbol{A} = \text{diag}\left([\alpha_1, \cdots, \alpha_n]\right)$. Hyper-prior distributions are also defined as $p(\alpha_i) = \text{Gamma}(a, b) = {}^1/_{\Gamma(a)}(b^a \alpha_i^{a-1} e^{-b\alpha_i})$ where $\Gamma(a)$ is the Gamma function [89]. This results in the joint distribution,

$$p(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x}) p\left(\boldsymbol{x}|\boldsymbol{\alpha}\right) p(\boldsymbol{\alpha}).$$

Following the variational approach, one can compute the factors of the approximating distribution one by one by taking the logarithm of the joint distribution shown above and then taking the expectation with respect to those variables not in the factor to be estimated. Applying this rule for $Q_{\boldsymbol{x}}(\boldsymbol{x})$,

$$\begin{aligned}
\log Q_{\boldsymbol{x}}(\boldsymbol{x}) &= \text{E}_{\boldsymbol{\alpha}}\left[\log p(\boldsymbol{y}|\boldsymbol{x}) + p\left(\boldsymbol{x}|\boldsymbol{\alpha}\right)\right] + C_1 \\
&= -\frac{1}{2} \boldsymbol{x}^T \left(\sigma^{-2} \boldsymbol{H}^T \boldsymbol{H} + \boldsymbol{A}\right) \boldsymbol{x} + \sigma^{-2} \boldsymbol{x}^T \boldsymbol{H}^T \boldsymbol{y} + C_1 \\
&= \log \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})
\end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{y} \\ \boldsymbol{\Sigma} &= \left(\sigma^{-2}\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{A}\right)^{-1}. \end{aligned} \tag{7.2}$$

Where matrix $\boldsymbol{A} = \text{diag}\left([\langle\alpha_1\rangle \cdots \langle\alpha_m\rangle]\right)$ is the diagonal matrix with $\langle\alpha_i\rangle = \text{E}(x)$ on its main diagonal. Constant C_1 gathers all the terms independent of $\boldsymbol{x}$ when computing $\log Q_{\boldsymbol{x}}(\boldsymbol{x})$.

Applying the same process for $Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$,

$$\begin{aligned} \log Q_{\alpha_i}(\alpha_i) &= \text{E}_{\boldsymbol{x},\alpha_{j\neq i}}\left[\log p(x_i|\alpha_i) + \log p(\alpha_i)\right] + C_2 \\ &= (a_i + 1/2 - 1)\log\alpha_i - (b_i + \langle x_i^2\rangle/2)\,\alpha_i + C_2 \\ &= \log\text{Gamma}(a_i + 1/2, b_i + \langle x_i^2\rangle/2) \\ &= \log\text{Gamma}(\tilde{a}_i, \tilde{b}_i). \end{aligned} \tag{7.3}$$

where

$$\begin{aligned} \tilde{a}_i &= a_i + \frac{1}{2} \\ \tilde{b}_i &= b_i + \frac{\langle x_i^2\rangle}{2}. \end{aligned} \tag{7.4}$$

Constant C_2 gathers all the terms independent of α_i that appear during the computation of $Q_{\alpha_i}(\alpha_i)$.

Based on the properties of the Gamma distribution and on very basic results for the Gaussian distribution it is possible to easily compute the following

$$\begin{aligned} \langle\boldsymbol{x}\rangle &= \boldsymbol{\mu} \\ \langle\boldsymbol{x}\boldsymbol{x}^T\rangle &= \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \\ \langle\alpha_i\rangle &= \frac{\tilde{a}_i}{\tilde{b}_i}. \end{aligned} \tag{7.5}$$

At this stage one recognises how the variational framework is more general than the Type-II maximum likelihood approach, acting in a fully Bayesian framework by providing closed form solutions to approximations of distributions that are intractable to derive analytically. The variational SBL (VSBL) is ideal for when the estimates of the hyper-prior distributions are required, something which becomes unmanageable within the *Type-II ML* framework. The price to pay of course is higher computational complexity since the algorithm iterates through computing Equations (7.4) and (7.2).

A quantity which is very useful to compute is the actual variational lower bound.

This is usually done to check for convergence. Indeed this will be the case with methods presented later in the text. This is given by the expression below

$$\mathcal{L} = \langle \log p(\boldsymbol{y}|\boldsymbol{x}) \rangle + \langle \log p(\boldsymbol{x}|\boldsymbol{\alpha}) \rangle + \langle \log p(\boldsymbol{\alpha}) \rangle - \langle \log Q(\boldsymbol{x}) \rangle - \langle \log Q(\boldsymbol{\alpha}) \rangle. \tag{7.6}$$

The individual quantities are computed as follows based on results for the Gaussian and the Gamma distribution [10],

$$\begin{aligned}
\langle \log p(\boldsymbol{y}|\boldsymbol{w}) \rangle &= \frac{n}{2}\log(2\pi) - \frac{\sigma^{-2}}{2}\sum_{i=1}^{n}\left[y_i^2 - 2y_i\langle \boldsymbol{x}\rangle^T \boldsymbol{h}_i + \boldsymbol{h}_i^T\langle \boldsymbol{x}\boldsymbol{x}^T\rangle \boldsymbol{h}_i\right] \\
\langle \log p(\boldsymbol{w}|\boldsymbol{\alpha}) \rangle &= -\frac{m}{2}\log(2\pi) + \frac{1}{2}\sum_{i=1}^{m}\left[\langle \log \alpha_i\rangle - \langle \alpha_i\rangle\langle x_i^2\rangle\right] \\
\langle \log p(\boldsymbol{\alpha}) \rangle &= m \cdot a_i \log b_i + (a_i - 1)\sum_{i=1}^{n}\left[\langle \log \alpha_i\rangle - b_i\langle \alpha_i\rangle\right] - m \cdot \log\Gamma(a_i) \\
\langle \log Q(\boldsymbol{x}) \rangle &= -m\frac{1+\log(2\pi)}{2} + \log\frac{|\boldsymbol{\Sigma}|}{2} \\
\langle \log Q(\boldsymbol{\alpha}) \rangle &= \sum_{i=1}^{n}\left[\tilde{a}_i \log \tilde{b}_i + (\tilde{\alpha}_i - 1)\langle \log \alpha_i\rangle - \tilde{b}_i\langle \alpha_i\rangle - \log\Gamma(\tilde{a}_i)\right]
\end{aligned}$$

7.3 The Fast Variational Sparse Bayesian Learning

A fast version of the variational algorithm was proposed in [86, 85] and is summarised in this section. The main goal of the authors is to establish a way of analytically detecting which components of $\boldsymbol{x}$ have reached a state where they can be *excluded* from the model, hence stop updating their corresponding variational distribution parameters. This is exactly what the case was in the Fast Type-II Maximum Likelihood (FMLM) in [90]. The authors manage to implement such a mechanism in a way similar to FMLM. As it turns out, for *uninformative prior distributions*, i.e., the fast VSBL (FVSBL) is equivalent to FMLM.

Starting off by substituting Equation (7.4) in the expression for the mean of the Gamma distribution,

$$\begin{aligned}
\tilde{\alpha}_i^{-1} = \langle \alpha_i^{-1}\rangle &= \frac{\tilde{b}_i}{\tilde{a}_i} \\
&= \frac{b_i + {}^1\!/_2(\mu_i^2 + \Sigma_{i,i})}{a_i + {}^1\!/_2} \\
&= \mu_i^2 + \Sigma_{i,i} \\
&= \boldsymbol{e}_i^T\left(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T\right)\boldsymbol{e}_i \\
&= \boldsymbol{e}_i^T\left(\boldsymbol{\Sigma} + \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{H}\boldsymbol{\Sigma}\right)\boldsymbol{e}_i
\end{aligned} \tag{7.7}$$

where in the third step it was assumed that the hyper-prior is *uninformative* thus setting $a_i = b_i = 0$. For computing $\langle x_i^2 \rangle$ the second expression from Equation (7.5) was used. Vector $\boldsymbol{e}_i$ denotes the i^{th} canonical vector of appropriate dimensions defined from the context.

The covariance matrix of the posterior in Equation (7.2) can be decomposed as,

$$\begin{aligned} \boldsymbol{\Sigma} &= \left(\sigma^{-2}\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{A}\right)^{-1} \\ &= \left(\sigma^{-2}\boldsymbol{H}^T\boldsymbol{H} + \tilde{\alpha}_i\boldsymbol{e}_i\boldsymbol{e}_i^T + \sum_{j\neq i}\tilde{\alpha}_j\boldsymbol{e}_j\boldsymbol{e}_j^T\right)^{-1} \\ &= \left(\boldsymbol{\Sigma}_{-i}^{-1} + \tilde{\alpha}_i\boldsymbol{e}_i\boldsymbol{e}_i^T\right)^{-1} \\ &= \boldsymbol{\Sigma}_{-i} - \frac{\boldsymbol{\Sigma}_{-i}\boldsymbol{e}_i\boldsymbol{e}_i^T\boldsymbol{\Sigma}_{-i}}{\tilde{\alpha}_i^{-1} + \boldsymbol{e}_i^T\boldsymbol{\Sigma}_{-1}\boldsymbol{e}_i} \end{aligned} \tag{7.8}$$

where matrix $\boldsymbol{\Sigma}_{-i}$ denotes the covariance matrix computed after excluding component indexed with i from the model.

Substituting Equation (7.8) in Equation (7.7) one arrives after some mathematical manipulations to the following implicit expression

$$\tilde{\alpha}_i^{-1} = w_i^2 + z_i - \frac{z_i^2 + 2z_iw_i^2}{\tilde{\alpha}_i^{-1} + z_i} + \frac{z_i^2w_i^2}{(\tilde{\alpha}_i^{-1} + z_i)^2}$$

where the following quantities have been defined

$$\begin{aligned} z_i &= \boldsymbol{e}_i^T\boldsymbol{\Sigma}_{-i}\boldsymbol{e}_i \\ w_i^2 &= \sigma^2\boldsymbol{e}_i^T\boldsymbol{\Sigma}_{-i}\boldsymbol{H}^T\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{H}\boldsymbol{\Sigma}_{-i}\boldsymbol{e}_i. \end{aligned} \tag{7.9}$$

Viewing the implicit equation as a recursive form the following map function is defined for iteration t,

$$\begin{aligned} \tilde{\alpha}_i^{[t+1]} &= \left[w_i^2 + z_i - \frac{z_i^2 + 2z_iw_i^2}{\frac{1}{\tilde{\alpha}_i^{[t]}} + z_i} + \frac{z_i^2w_i^2}{(\frac{1}{\tilde{\alpha}_i^{[t]}} + z_i)^2}\right]^{-1} \\ &= F(\tilde{\alpha}_i^{[t]}) \end{aligned} \tag{7.10}$$

which gives the value of $\tilde{\alpha}_i^{[t+1]}$ at iteration t given its value at the previous iteration.

The authors then adopt a fixed point analysis by letting $t \to \infty$ and study the fixed point of map function F for stability. At convergence it holds that $\tilde{\alpha}_i^{[t+1]} = \tilde{\alpha}_i^{[t]} = \tilde{\alpha}_i^*$. The two fixed points are found which are asymptotically stable:

$$\tilde{\alpha}_i = \begin{cases} (w_i^2 - z_i)^{-1}, & \text{if } w_i^2 > z_i \\ +\infty, & \text{if } w_i^2 \leq z_i. \end{cases} \tag{7.11}$$

The first fixed point is found by solving $\tilde{\alpha}_i^* - F(\tilde{\alpha}_i^*) = 0$ while the second one is readily available from inspection. The authors then compute the absolute value of the derivative of the map function at the first fixed point,

$$\left.\frac{dF(\tilde{\alpha}_i^*)}{d\tilde{\alpha}_i^*}\right|_{\tilde{\alpha}_i=(w_i^2-z_i)^{-1}} = -\frac{z_i(z_i-2w_i^2)}{w_i^4} \tag{7.12}$$
$$\left|\frac{z_i(z_i-2w_i^2)}{w_i^4}\right| < 1$$

In order for the fixed point $\tilde{\alpha}_i = (w_i^2 - z_i)^{-1}$ to be *asymptotically stable and positive* (absolute value of the derivative at that point smaller than 1) the rule $w_i^2 > z_i$ must hold, otherwise the fixed point diverges and $\tilde{\alpha}_i = +\infty$. The positivity constraint rises from the definition of the Gamma distribution.

It can be seen from Equation (7.12) that the intermediate iterations for computing the model parameters can be circumvented by iteratively computing the fixed points. Moreover by studying the stability of these fixed points it becomes possible to *analytically* prune those parameters for which the fixed point is not asymptotically stable. It is uncertain whether the convergence points of the two approaches (the traditional and the fast) will converge be the same. Nevertheless this is quite practical since for sparse signal recovery a threshold function is usually required. The pruning rule takes this part. This results in a highly efficient variational algorithm for sparse Bayesian learning.

7.3.1 Equivalence with Type-II Maximum Likelihood

From Chapter 5 the optimal values for the variance point estimates where found via Type-II Maximum Likelihood to be,

$$\alpha_i = \begin{cases} \frac{s_i^2}{q_i^2 - s_i}, & \text{for } q_i^2 > s_i \\ +\infty, & \text{for } q_i^2 \leq s_i. \end{cases}$$

where $s_i = \boldsymbol{h}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{h}_i, q_i = \boldsymbol{h}_i^T \boldsymbol{C}_{-i}^{-1} \boldsymbol{y}$ and $\boldsymbol{C}_{-i} = \left(\sigma^2 \boldsymbol{I} + \sum_{j \neq i} \alpha_j^{-1} \boldsymbol{h}_j \boldsymbol{h}_j^T\right)$.

It is easy to show by computing $\boldsymbol{C}_{-i}^{-1}$ that the following equalities hold [86]

$$z_i = s_i^{-1}$$
$$w_i^2 = \frac{q_i^2}{s_i^2}.$$

This summarises to the fact that for an *uninformative* prior Gamma distribution the pruning rules and the variance values α_i derived from the Fast variational approach are in fact the same as the ones given by the Fast Marginal Maximum

Likelihood.

7.4 Employing an Informative Prior

It was underscored in Section 7.2 that the VSBL approach allows for estimating not only point estimates for α_i but the parameters of its distribution hence providing a fully Bayesian framework. It was also shown above that by iterating Equations (7.2) and (7.4) the variational lower bound is maximised and a solution which best describes the data in $\boldsymbol{y}$ will be recovered. Focusing on the properties of the hierarchical model of SBL the fast Type-II (FMLM) and the fast Variational (FVSBL) algorithms manage to achieve excellent performance and scalability by assuming an *uninformative* prior distribution for α_i. It was also shown [86] that both algorithms are effectively equivalent regarding the analytical pruning rules and solutions.

Reverting to the initial reasons for adopting the variational approach; one wishes to place different preference over the variance of each x_i. This is mostly attributed to relevant experience for the given sparse problem or due to some expert system or empirical model. Put simply, the variance of the sparse components might have a certain bias towards some values than some others. An example which will occupy a whole chapter later in the text is that of sparse channel estimation. Envision a component x_i of a multipath communications channel residing at delay i and another component x_j. Assume that $j > i$, i.e., component x_j resides at a greater distance from the receiver than x_i. If one adopts the known hierarchical Bayesian model then it is expected that the variances are likely to satisfy $\alpha_i > \alpha_j$ due to the physical phenomena governing the channel, i.e, components closer to the receiver will have exponentially larger magnitudes. It is desired that this *prior knowledge* about the nature of the problem to be somehow implemented into the inference algorithm.

The variational approach allows for this to be implemented naturally since for each α_i a complete distribution is assumed and a certain statistical bias can be imposed by appropriately selecting the corresponding values for $a_i > 0, b_i > 0$ (the parameters of the hyper-prior Gamma distribution). It was highlighted in Section 7.1 that Type-II is not suitable for this task. Even though the VSBL algorithm can handle this addition it has been shown that it is not efficient for sparse signals.

Attempting to use the FVSBL algorithm presented in the previous section halts at the second line of Equation (7.8) for the update of the mean of α_i,

$$\tilde{\alpha}_i^{-1} = \frac{b_i + {}^1\!/_2(\mu_i^2 + \Sigma_{i,i})}{a_i + {}^1\!/_2}.$$

By setting non-zero values for a_i and b_i one quickly realises that there is a major discrepancy with the analytical pruning rule in Equation 7.11 since the fixed point

of $\tilde{\alpha}_i = +\infty$ can no longer be reached. Strictly speaking the FVSBL can still be employed to perform inference but the performance gains are questionable since no pruning will be taking place for any component. In such case for *an informative prior* the original variational approach is more fitting. It will be demonstrated later in the text - as another contribution of this thesis - that there exists a set of modifications that actually allow for a fast algorithm to exist for an informative variational SBL model.

7.4.1 Related Approaches

Strictly speaking the problem of infusing prior information in the recovery of a sparse signal can be addressed in a deterministic way in a simplistic setting. Consider a sparse signal $\boldsymbol{x}$ with support set $\mathcal{T}$. A sparse recovery algorithm is used to produce an estimate $\hat{\boldsymbol{x}}$. For the sake of discussion assume that the Orthogonal Matching Pursuit algorithm is used (see Chapter 2, Algorithm 1). The algorithm is usually initialised with an empty support set $\mathcal{T}^{[0]}$ and a $\hat{\boldsymbol{x}} = \mathbf{0}$. It is possible that some estimate $\tilde{\mathcal{T}}$ of the true support set is available along with some prior estimate $\tilde{\boldsymbol{x}}$. Quite simply the OMP algorithm can be initialised with $\tilde{\mathcal{T}}$ and $\tilde{\boldsymbol{x}}$ instead of the usual initialisations. This of course affects convergence and recovery performance in numerous ways. To be pedantic it is also related to the problem of dynamic sparse signal recovery discussed in Chapter 6 and more specifically to that of sparse recovery problems with partially known support. It is easy to envision that the available estimates $\tilde{\mathcal{T}}, \tilde{\boldsymbol{x}}$ might actually be seen as the output of the algorithm from one iteration to the next. The authors in [102] have addressed this issue and have been able to derive certain performance guarantees.

Despite this vantage point is quite interesting it will not occupy this text any further. The approaches studied here despite being applied for sparse recovery or basis selection, they differ from non-Bayesian methods since they attempt to fit the given measurements in a certain hierarchical Bayesian model. From what has been discussed so far this involves the definition of a hyper-prior, a prior distribution and a measurement distribution. Sparsity is then realised via an inference mechanism which provides estimates for the model parameters. In this rationale, any prior knowledge about the sparse signal has to be implemented in a systematic way that respects the model structure so that meaningful results are produced.

It is a fact that not many cases exist in the bibliography that explicitly address the problem of adopting *informative* prior distributions for the specific hierarchical Bayesian model. From a completely different perspective but still nearby the authors in [64] perform a study for the benefits of informative prior distributions for inference in graphical models. A very vague mention for sparsity is made in [43] where a very

specific prior is proposed for general regression problems. The authors propose a prior that is able to cope with large datasets in which the number of predictors is small. This is related to sparse system models but the overall proposed structure does not fit in the cases studied here. A discussion of similar nature takes place in [70] but for single-layer graphical models which is clearly more confined than the sparse Bayesian model considered here.

7.5 Modified Fast Variational Bayesian Learning

In this section a set of modifications to the fast VSBL approach that are an original contribution of the thesis. The same analysis is carried out as before like in [86] but this time an informative prior is employed so one has $a_i > 0, b_i > 0$. A similar map function like the one in Equation (7.10) is derived,

$$\begin{aligned}\frac{(2a_i+1)}{\tilde{\alpha}_i^{[t+1]}} &= \left[2b_i + w_i^2 + z_i - \frac{z_i^2 + 2w_i^2 z_i}{\frac{1}{\tilde{\alpha}_i^{[t]}} + z_i} + \frac{z_i^2 w_i^2}{(\frac{1}{\tilde{\alpha}_i^{[t]}} + z_i)^2}\right] \\ &= G(\tilde{\alpha}_i^{[t]}). \end{aligned} \tag{7.13}$$

Likewise, to derive the fixed points of the above map function the roots of the following equation must be found,

$$\tilde{\alpha}_i^* - G\left(\tilde{\alpha}_i^*\right) = 0. \tag{7.14}$$

It turns out that the polynomial in Equation (7.14) is not as *well-posed* as the one in Equation (7.10). A quick inspection reveals that the additional terms attributed to the informative prior cause this issue and map function G cannot be factorised easily enough to allow for simple closed form solutions for the fixed points. A study in [12] has pointed out these issues as well. The solutions of this cubic polynomial of course can be derived using basic results from algebra; the expressions for the roots of a cubic can be analytically produced but their intricacy does not allow for any results like the ones derived earlier. The problem of choosing one of the three roots of the polynomial then arises.

Theorem 15 (Karseras, Dai)**.** *Assume a Gamma hyper-prior for α_i with $a_i > 0, b_i > 0$ and that the values of $\alpha_{j\neq i}$ are fixed. Let $\tilde{\alpha}_i^*$ be the value at which the map function G converges when iteration $t \to \infty$. Then $\beta_i^* = \frac{1}{\tilde{\alpha}_i^*} + z_i$ is one of the three solutions of the cubic polynomial,*

$$f(\beta_i^*) + g(\beta_i^*) = 0 \tag{7.15}$$

that satisfies $\beta_i^* > z_i$*, where*

$$f(\beta_i^*) = \left(\beta_i^* - w_i^2\right)\left(\beta_i^* - z_i\right)^2$$
$$g(\beta_i^*) = 2(\beta_i^*)^2(a_i(\beta_i^* - z_i) - b_i).$$

Proof. Starting from Equation (7.13) and substituting the fixed point $\tilde{\alpha}_i^* = \tilde{\alpha}_i^{[t+1]} = \tilde{\alpha}_i^{[t]}$,

$$(\tilde{\alpha}_i^*)^{-1} = \frac{1}{2a_i + 1}\left[2b_i + w_i^2 + z_i - \frac{z_i^2 + 2w_i^2 z_i}{(\tilde{\alpha}_i^*)^{-1} + z_i} + \frac{z_i^2 w_i^2}{\left((\tilde{\alpha}_i^*)^{-1} + z_i\right)^2}\right]$$
$$(\tilde{\alpha}_i^*)^{-1} + z_i = \frac{1}{2a_i + 1}\left[2b_i + w_i^2 + z_i - \frac{z_i^2 + 2w_i^2 z_i}{(\tilde{\alpha}_i^*)^{-1} + z_i} + \frac{z_i^2 w_i^2}{\left((\tilde{\alpha}_i^*)^{-1} + z_i\right)^2}\right] + z_i$$
$$(\beta_i^*)^3 - (\beta_i^*)^2(w_i^2 + 2z_i) + \beta_i^*(z_i^2 + 2z_i w_i^2) - z_i^2 w_i^2 = (\beta_i^*)^2(2b_i - 2a_i(\beta_i^* - z_i))$$
$$\underbrace{(\beta_i^* - w_i^2)(\beta_i^* - z_i)^2}_{f(\beta_i^*)} + \underbrace{2(\beta_i^*)^2(a_i(\beta_i^* - z_i) - b_i)}_{g(\beta_i^*)} = 0.$$

At the second step the substitution took place, $\beta_i^* = (\tilde{\alpha}_i^*)^{-1} + z_i$. One arrives at the claimed result after a careful re-ordering of the terms. □

Theorem 15 dictates that the map function is indeed quite simple to comprehend; it can be seen as being consisted of two discrete parts. Using variable β_i as a proxy the situation becomes easier to handle. As will be shown next this is the first step towards arriving at some positive results about the aforementioned problems. The first part, function $f(\beta_i^*)$ is actually the same as the map function for when an uninformative prior is used. The second part, function $g(\beta_i^*)$ represents that bias induced by the additional terms for when $a_i > 0, b_i > 0$. It can be easily verified that for the uninformative case function g vanishes. Solving $f(\beta_i^*) = 0$ will result in the same fixed points for FVSBL.

7.5.1 A set of Practical Rules

Theorem 15 provides useful intuition for the fixed points and facilitates further qualitative analysis by using functions f and g. Proposition 1 suggests that it is possible to identify the cardinality of the fixed set given certain conditions. From basic algebra it is known that a cubic polynomial can have either 3 real roots or 1 real root and two imaginary ones (conjugate to one another). From this point on, *valid roots* are those that belong in $\mathbb{R}^+$ since the according to the model the roots represent variances which are positive dimensionless numbers.

Proposition 1 (Karseras, Dai)**.** *Assume that the same conditions hold as in Theorem 15. Solving Equation* (7.14) *can result in one of the following cases:*

1. *if* $w_i^2 \leq z_i$*, then there* may *exist three distinct valid roots,*
2. *if* $w_i^2 > z_i$*, then for* $\frac{b_i}{a_i} \geq \frac{2}{3}(w_i^2 - z_i)$ *there exists* only one *valid root,*
3. *if* $w_i^2 > z_i$*, then for* $\frac{b_i}{a_i} < \frac{2}{3}(w_i^2 - z_i)$ *there* may *exist three distinct valid roots.*

Proof. The process starts from computing the roots of the two auxiliary functions $f(\beta_i)$ and $g(\beta_i)$ and of their derivatives. For $f(\beta_i)$

$$\begin{aligned} f(\beta_i) &= (\beta_i - w_i^2)(\beta_i - z_i)^2 = 0 \\ &\Rightarrow \quad \underline{\beta_i = w_i^2 \text{ or } \beta_i = z_i} \\ f'(\beta_i) &= (\beta_i - z_i)\left[(\beta_i - z_i) + 2(\beta_i - w_i^2)\right] = 0 \\ &\Rightarrow \quad \underline{\beta_i = z_i \text{ or } \beta_i = {}^1\!/\!_3(2w_i^2 + z_i)}. \end{aligned}$$

For $g(\beta_i)$

$$\begin{aligned} g(\beta_i) &= 2(\beta_i)^2(a_i(\beta_i - z_i) - b_i) = 0 \\ &\Rightarrow \quad \underline{\beta_i = 0 \text{ or } \beta_i = {}^{b_i}\!/\!_{a_i} + z_i} \\ g'(\beta_i) &= 6a_i\beta_i^2 - 4\beta_i(a_i z_i + b_i) = 0 \\ &\Rightarrow \quad \underline{\beta_i = 0 \text{ or } \beta_i = {}^2\!/\!_3({}^{b_i}\!/\!_{a_i} + z_i)} \end{aligned}$$

A sign diagram is drawn for both functions for the case where $w_i^2 < z_i$.

β_i	0		w_i^2		$\frac{2w_i^2+z_i}{3}$		z_i		$+\infty$
$f'(\beta_i)$		$+$		$+$	0	$-$	0	$+$	
$f(\beta_i)$		$-$	0	$+$		$+$	0	$+$	

β_i	0		$2({}^{b_i}\!/\!_{a_i} + z_i)/3$		${}^{b_i}\!/\!_{a_i} + z_i$		$+\infty$
$g'(\beta_i)$	0	$-$	0	$+$		$+$	
$g(\beta_i)$	0	$-$		$-$	0	$+$	

From the sign table for f it is directly evident that there is a maximum at $\beta_{max} = {}^{(2w_i^2 + z_i)}\!/\!_3$ and a minimum at $\beta_{min} = z_i$. Function g exhibits a root at ${}^{b_i}\!/\!_{a_i} + z_i$. Based on this it is certain that there exists a root $\beta_i^1 \in (z_i, {}^{b_i}\!/\!_{a_i} + z_i)$ for

the sum $f+g$. This happens since the point

$$f(\beta_{min}) + g(\beta_{min}) < 0$$

will always be negative hence $f+g$ will exhibit a sign change in the interval $(z_i, {}^{b_i}/_{a_i} + z_i)$.

It is possible to identify the existence of maybe two additional roots by observing the behaviour of f and g in the interval (w_i^2, z_i). More specifically, in case,

$$f\left(\frac{2w_i^2 + z_i}{3}\right) + g\left(\frac{2w_i^2 + z_i}{3}\right) > 0$$

then there will certainly exist another two roots $\beta_i^2, \beta_i^3 \in (w_i^2, z_i^2)$ since $f(w_i^2) + g(w_i^2) < 0$ and $f(z_i) + g(z_i) < 0$ always. It is easy to see that in case,

$$f\left(\frac{2w_i^2 + z_i}{3}\right) + g\left(\frac{2w_i^2 + z_i}{3}\right) = 0$$

then there will only one additional root and $\frac{2w_i^2+z_i}{3}$ will be a local maximum of $f+g$. Lastly, in case

$$f\left(\frac{2w_i^2 + z_i}{3}\right) + g\left(\frac{2w_i^2 + z_i}{3}\right) < 0$$

there will be no additional roots. In all cases if any additional roots exist they will lie in the interval (w_i^2, z_i).

The same is performed for the case when $w_i^2 = z_i$. This case being easier the derivative test is not required.

β_i	0	$w_i^2 = z_i$	${}^{b_i}/_{a_i} + z_i$	$+\infty$
$f(\beta_i)$	$-$	0	$+$	$+$
$g(\beta_i)$	$-$	$-$	0	$+$

For this case it is easy to verify that

$$g(w_i^2 = z_i) < 0$$

and that

$$f\left({}^{b_i}/_{a_i} + z_i\right) > 0.$$

By inspecting the table it is then evident that $f+g$ will exhibit a change of signs in the interval $(w_i^2 = z_i, {}^{b_i}/_{a_i} + z_i)$ indicating the existence of only one solution.

The same is repeated for the case when $w_i^2 > z_i$.

β_i	0		z_i		$\frac{(2w_i^2+z_i)}{3}$		w_i^2		$+\infty$
$f'(\beta_i)$		$+$	0	$-$	0	$+$		$+$	
$f(\beta_i)$		$-$	0	$-$		$-$	0	$+$	

β_i	0		$2(b_i/a_i + z_i)/3$		$b_i/a_i + z_i$		$+\infty$
$g'(\beta_i)$	0	$-$	0	$+$		$+$	
$g(\beta_i)$	0	$-$		$-$	0	$+$	

Actually the sign table for $g(\beta_i)$ remains the same but is repeated for convenience. From the table f exhibits a minimum at $\beta_{min} = \frac{(2w_i^2+z_i)}{3}$ and a root at w_i^2. Since g has a root at $b_i/a_i + z_i$ the existence of more than one roots depends on the relationship between β_{min} and $b_i/a_i + z_i$.

More specifically, if $b_i/a_i + z_i \geq (2w_i^2 + z_i)/3 \Rightarrow b_i/a_i \geq 2(w_i^2 - z_i)/3$, one identifies three cases for the *single root* β_i^1

$$\begin{aligned}
&\text{if } w_i^2 > b_i/a_i + z_i \text{ then } \beta_i^1 \in \left(b_i/a_i + z_i, w_i^2\right), \\
&\text{if } w_i^2 = b_i/a_i + z_i \text{ then } \beta_i^1 = b_i/a_i + z_i, \\
&\text{if } w_i^2 < b_i/a_i + z_i \text{ then } \beta_i^1 \in \left(w_i^2, b_i/a_i + z_i\right).
\end{aligned}$$

In the first and third cases there is a sign change in the corresponding intervals while for the case in the middle the result is immediately drawn from the table since both f and g exhibit a root at the same point. There only exists a single root since $f+g$ exhibits only a single sign change.

Now focus is turned at the case where $b_i/a_i + z_i < (2w_i^2 + z_i)/3 \Rightarrow b_i/a_i < 2(w_i^2 - z_i)/3$. By inspecting the tables for the interval $(z_i, b_i/a_i + z_i,)$ one notices that $f+g < 0$ and that two sign changes are possible depending on the values of $g(\beta_i)$ for $\beta_i > b_i/a_i + z_i$ and $f(\beta_{min})$. One then can deduce that three distinct real roots are possible in the interval $(b_i/a_i + z_i, w_i^2)$. □

Proposition 1 suggests that in some cases it is possible for the fixed set to contain three *distinct real fixed points* without being able to identify which one should be selected. Comparing the results from the proposition and the pruning rule in Equation (7.11) one quickly notices that indeed it is not applicable as in some cases some basis functions might get pruned when they should not and vice versa. Indeed this agrees with intuition as well since from what was demonstrated above the existence or not of more than one valid fixed points depends on the value of a_i, b_i and more specifically on the relation ship between the local extrema of functions

f and g. The only situation in which no selection needs to take place is in case 2 where a single valid fixed point can exist. A numerical example for the third case can be constructed by setting $w_i^2 = 0.6, z_i = 0.4, a_i = 0.02$ and $b_i = 0.002$. Similar examples can be constructed for all cases.

Fixed point selection

Focusing on those cases where there exist three distinct possible fixed points for $\tilde{\alpha}_i^*$ (equivalently for β_i^*), a choice must be made. In order to resort to a choice the stability of each fixed point has to be assessed like in Equation (7.12). This requires the analytical expression of $\frac{dG}{d\tilde{\alpha}_i}$ to be calculated at each of the three possible fixed points $\tilde{\alpha}_i^*$ and then to check whether they are asymptotically stable or not. The highly complicated expressions unfortunately prohibit this analysis and a work around must be found.

Since the variational approach aims to maximise the variational lower bound a reasonable choice is to choose the fixed point that achieves the variational lower bound to increase the most. This would effectively mean that the update equations for the parameters of the variational model to be updated in triplicate and then for a choice to be made based on the variational lower bound (also computed three times). Since this would lead to an increase in the computational demands, especially for large systems and sparse signals a different route is chosen.

The following conjecture which suggests a simple and effective remedy to this problem.

Conjecture 1 (Karseras, Dai)**.** *In case the solution of Equation* (7.14) *results in three distinct real roots* $0 < \tilde{\alpha}_i^1 < \tilde{\alpha}_i^2 < \tilde{\alpha}_i^3 < +\infty$ *then* $\tilde{\alpha}_i^1$ *causes the variational lower bound* $\mathcal{L}$ *to increase the most. The same holds if two distinct real roots exist.*

In Conjecture 1 it is argued that in the case where three distinct real fixed points exist, the best choice as far as the lower bound is concerned is to select the smallest one in value. The motivation behind the conjecture is that since the hierarchical model tends to promote sparse signals then choosing the fixed point which corresponds to the smallest variance $\tilde{\alpha}_i$ is more likely to contribute towards inferring a sparse x_i , hoping that this way the approximating distribution $Q(\boldsymbol{x}, \boldsymbol{\alpha})$ will be closer to the truth. The conjecture has not been able to be proven yet but empirical results show that it achieves very good results.

7.5.2 Controlling complexity for superfluous parameters

A practical way to allow further control over the overall complexity is to update only those parameters for which $\tilde{\alpha}_i^*$ is above a certain threshold. In [86] it was

discussed that

$$\frac{w_i^2}{z_i} = SNR_i$$

can be seen as an estimate of the signal-to-noise ratio for component x_i. Then the pruning rule can be recast as

$$w_i^2 > z_i \cdot SNR_i'$$

for a given SNR_i' value. This proposal is perfectly in sync with the proposed approaches for an informative prior. Quite easily one can chose to update only those parameters for which $\tilde{\alpha}_i^* > \sigma^2$, i.e., that exhibit variance greater of that of the noise.

Algorithm 8 Extended Fast Variational Sparse Bayesian Learning

Input: $\boldsymbol{H}$, $\boldsymbol{y}, \sigma^2$, hyper-prior parameters $a_i, b_i \forall i \in [1, m]$ and threshold τ.
Initialise:

1. Initialise σ^2 to some appropriate value and $\tilde{\alpha}_i^{[-1]} = +\infty$ for $i \in [1, m]$.
2. Compute $\tilde{\alpha}_i^{[0]}$ from Equation (7.13).
3. $\mathcal{T} = \{$ index i for which α_i is minimum $\}$.
4. Compute $\boldsymbol{\Sigma}, \boldsymbol{\mu}$ and $\tilde{a_i}, \tilde{b_i}$ using only the indices $i \in \mathcal{T}$.

Iteration[t]:

1. For each $i \in \mathcal{T}^c = [1, m] - \mathcal{T}$:
   - Compute w_i^2, z_i according to Equations (7.9).
   - Compute solutions of Equation (7.14) using a numerical method and form fixed set $\mathcal{A}$. The fixed set will contain three elements.
   - Set $\tilde{\alpha}_i^{[t]} = \min\{\mathcal{A}\}$ where $\min\{\mathcal{A}\}$ selects the *minimum positive real* element from $\mathcal{A}$.
   - If $\tilde{\alpha}_i^{[t]} < \tau$ then set $\mathcal{T} = \{i, \mathcal{T}\}$ and update $\boldsymbol{\Sigma}, \boldsymbol{\mu}$ based on the new $\tilde{\alpha}_i^{[t]}$.
2. For each $i \in \mathcal{T}$:
   - Compute w_i^2, z_i according to Equations (7.9).
   - Compute solutions of Equation (7.14) using a numerical method and form the fixed set $\mathcal{A}$.
   - Set $\tilde{\alpha}_i^{[t]} = \min\{\mathcal{A}\}$ where $\min\{\mathcal{A}\}$ selects the *minimum positive real* element from $\mathcal{A}$.
   - If $\tilde{\alpha}_i^{[t]} > \tau$ then set $\tilde{\alpha}_i^{[t]} = +\infty$.
   - Update values of $\boldsymbol{\Sigma}, \boldsymbol{\mu}$ based on the new $\tilde{\alpha}_i^{[t]}$.
3. Compute the variational lower bound $\mathcal{L}^{[t]}$ according to Equation (7.6). If the change form $\mathcal{L}^{[t-1]}$ is below some threshold then quit.

Output:

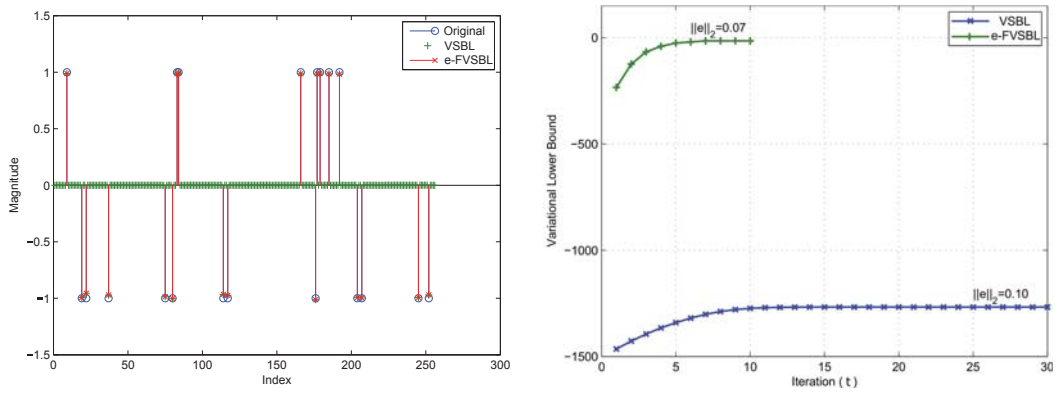
1. Estimated support set $\mathcal{T}$ and sparse signal $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

7.5.3 The Extended Fast Variational Sparse Bayesian Learning Algorithm

The steps of the extended fast variational algorithm are described in Algorithm 8. The reader will notice that the steps of the algorithm are divided into two parts; the first part being responsible for updating the statistics for the indices outside of a support set while the second part being responsible for updating those elements corresponding to indices inside a support set. This is done to facilitate the use of a threshold τ like it was discussed in Subsection 7.5.2. This is of course optional and in case a threshold is not used the algorithm would have to be altered slightly (basically one then would set $\mathcal{T} = [1, m]$ constantly and the first group of updates would vanish).

Also importantly the algorithm requires the constant updating of the quantities $\boldsymbol{\Sigma}, \boldsymbol{\mu}$ of the posterior. This can be efficiently performed by extensive use of the Woodbury matrix identity for adding, deleting and updating of specific parts of a matrix or a vector. The expressions for such efficient operations can be found in [86] but in [90] as well.

7.6 Empirical Results



(a) *e-FVSBL returns an* exactly sparse *signal in contrast to VSBL.*

(b) *e-FVSBL converges faster, at a higher lower bound with smaller error.*

Figure 7.2: *Reconstruction performance for* $\boldsymbol{H} \in \mathbb{R}^{128\times256}$, $|\mathcal{T}| = 20$ *and* $\sigma^2 = 0.01$ *for a* zero-one *sparse signal. The Gamma distribution parameters are set to* $a_i = b_i = 0.1^3$.

The performance of the proposed algorithm namely the extended Fast Variational Sparse Bayesian Learning e-FVSBL is assessed. At first the performance of e-FVSBL is compared against VSBL in terms of sparse signal recovery, convergence speed (iteration count t) and reconstruction error $\boldsymbol{e}$. The entries of $\boldsymbol{H} \in \mathbb{R}^{128\times256}$ are drawn from $\mathcal{N}(0, 1/\sqrt{n})$. Signal $\boldsymbol{x}$ is a *zero-one* sparse signal with support set $\mathcal{T}$ chosen uniformly at random from $[1, m]$ with $|\mathcal{T}| = 20$ and $\sigma^2 = 0.01$. A single run

of the algorithms was performed with $\boldsymbol{y}$, σ^2 as input. The sparsity level needs not be known as was discussed in earlier chapters. The Gamma parameters were set to $a_i = b_i = 0.1^3$ for all $i \in [1, m]$. A threshold $\tau = \sigma^2$ was also used.

In Figure 7.2a the original signal is shown versus the recovered. The e-FVSBL does not suffer from the small amplitude components giving an exactly sparse signal. This causes a decrease in convergence speed as shown in Figure 7.2b where e-FVSBL converges in only 10 iterations. Convergence was assumed when the difference in the variational lower bound went below 0.1^8. The number of iterations was limited to 30 since VSBL during tests took more than 700 iterations to converge. It is also shown that the e-FVSBL achieves a significantly higher variational lower bound and higher reconstruction accuracy.

Problem	Iterations (t)		Runtime (sec)	
Size	e-FVSBL	VSBL	e-FVSBL	VSBL
128×256	10	40	0.44	0.75
256×512	9	39	1.20	3.91
512×1024	8	38	6.63	25.28
1024×2048	9	38	62.2	142.88

Table 7.1: *Comparison for $a_i = b_i = 0.1^3$, $|\mathcal{T}| = 20$, $\sigma^2 = 0.01$ and for increasing problem size.*

Table 7.1 compares the convergence speed and runtime (in seconds) of e-FVSBL. In this scenario increasing problem sizes are considered, i.e., the design matrix $\boldsymbol{H}$ is re-sampled at different sizes. The sparsity level and prior distribution strength are kept unchanged. It is evident that the proposed algorithm succeeds in recovering sparse signals under the *informative* assumption showcasing significantly reduced computational complexity and runtime.

Prior	$\lvert\mathcal{S}\rvert = 15$			$\lvert\mathcal{S}\rvert = 30$		
$a_i = b_i$	$\lVert\boldsymbol{e}\rVert_2$	$\lvert\mathcal{T}'\rvert$	t	$\lVert\boldsymbol{e}\rVert_2$	$\lvert\mathcal{T}'\rvert$	t
0.1^2	0.54	71	35	0.13	50	11
1	0.57	71	48	0.13	50	10
10^2	0.55	69	24	0.12	50	9
10^5	0.62	69	27	0.12	50	9

Table 7.2: *Comparison for $\boldsymbol{H} \in \mathbb{R}^{128\times256}$, $|\mathcal{T}| = 50$, $\sigma^2 = 0.01$ at different sizes of $\mathcal{S}$ against different prior strength.*

For Table 7.2 a stringent scenario is assumed. For a subset $\mathcal{S} \subset \mathcal{T}$ a stronger prior is employed expressing prior preference. For subset $i \in \mathcal{T} - \mathcal{S}$ the prior is set to $a_i = b_i = 0.1^5$ while the prior for $i \in \mathcal{S}$ varies as shown in Table 7.2. The

algorithm is tested for different sizes of $\mathcal{S}$ against reconstruction error, recovered support set cardinality $\|\mathcal{T}'\|$ and iteration count. It is considered that $|\mathcal{T}| = 50$ while $\boldsymbol{H} \in \mathbb{R}^{128\times256}$. Recovery using *uninformative* priors under-performs with $\|\boldsymbol{e}\|_2 = 0.55$ and $|\mathcal{T}'| = 71$. Table 7.2 shows that for adequately large $\mathcal{S}$, i.e. adequate prior information, exact recovery is possible. By increasing the strength of the prior is it also possible to improve convergence speed.

7.7 Conclusion

Certain cases exist where the hierarchical model for sparse Bayesian learning is crippled when it comes to adopting some attributes of the system under study. In cases like this it is presumed that some knowledge is available for the sparse components of a signal, for example some components should exhibit a smaller magnitude than others because of the physical properties of the problem.

Under the usual assumptions for uninformative hyper-prior distributions that are made in the bibliography the available inference algorithms are capable of excellent performance. It was underscored that the nice properties of these algorithms vanish when it comes to the cases of interest described above. The main contributions in this chapter is that informative hyper-prior distributions can be used to inject certain prior statistical knowledge in the model and that efficient algorithms can be derived to perform inference.

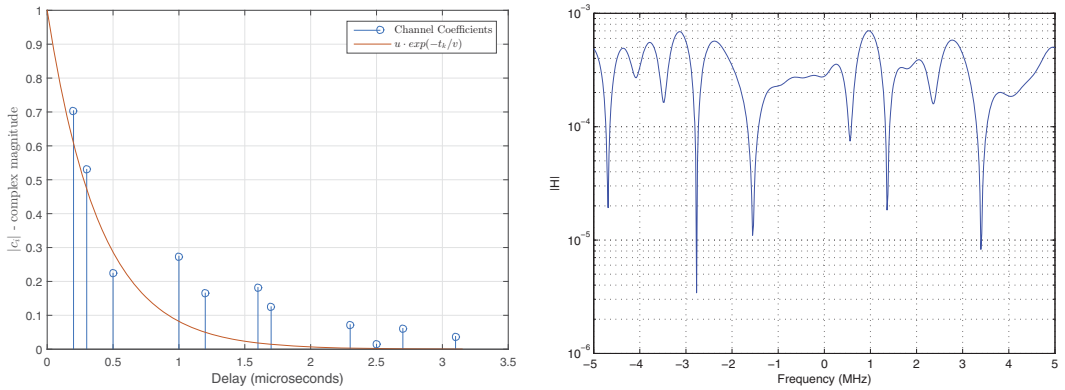
Chapter 8

Multipath Channel Estimation

This chapter builds upon the idea of incorporating prior statistical information for the problem of channel equalisation. More specifically pilot-assisted Orthogonal Frequency-Division Multiplexing (OFDM) systems are considered. Focus is on cases where prior information about the distribution of the channel coefficients can be used to enhance the equalisation process and achieve improved performance and convergence speed. This is performed by considering certain *informative prior distributions* for the channel coefficients c_i.

Assuming a sparse multipath channel, the equalisation problem is amenable to a Bayesian formulation and inference can be performed in the well-known Sparse Bayesian Learning (SBL) framework which was discussed extensively in previous chapters. This translates to adopting a hyper-prior $p(\alpha_i)$ for the variance of each coefficient $p(c_i|\alpha_i)$ which will be able to express certain preference over some values of α_i. The variational algorithms that where analysed earlier in the text are indeed a perfect fit for this task. It was shown that the previously proposed Fast Variational SBL (FVSBL) algorithm is capable of efficient inference in a true Bayesian setting but only in the case of *uninformative prior distributions*. In Chapter 7 a refined analysis provided a set of very practical *extensions* to mitigate these problems with the FVSBL approach.

The previously proposed algorithm, namely the extended Fast Variational Sparse Bayesian Learning (e-FVSBL) is adapted to perform equalisation for multipath fading channels in OFDM systems. The proposed approach shows how to exploit prior knowledge about exponentially decaying power-delay channel profiles to produce accurate estimates and improve convergence. Empirical results are presented for synthetic test cases. A real-world example is also presented with a dataset acquired from private transmissions tests at the Tsinghua University, China.



(a) *Channel coefficients with an exponentially-decaying power delay profile.*

(b) *The corresponding channel frequency response.*

Figure 8.1: *Example of a simulated multipath fading channel according to the model described by Equations* (8.1) *and* (8.2).

8.1 Multipath Fading Transmission Channels

Many wireless channels display a scattering nature which results in a *sparse* impulse response, i.e., it can be seen as a finite sum of impulses over a finite time duration. Usually, the power-delay profile of the channel exhibits an exponential decay of the multipath components. This phenomenon can be attributed to the propagation of electromagnetic waves and its explanation falls outside the purpose of this text. For the discussion to follow it will be used as a fact but the methods developed are not limited to such an assumption and accept any class of channel profiles.

Envision a sparse multipath channel described by

$$c(t) = \sum_{k=1}^{K} c_k \delta(t - t_k), \quad 0 \leq t_k \leq T_{cp}, \tag{8.1}$$

a finite sum of Dirac impulses with complex coefficients c_k. The delay values t_k are chosen uniformly at random from the continuous interval $[0, T_{cp}]$ where T_{cp} in reality is equal to $+\infty$. For practical purposes a finite value is considered to aid equalisation and in OFDM systems it is usually considered to be less than the duration of the cyclic prefix. A valid choice for K is the one proposed in [74] where

$$p(K) = \text{Pois}(K|M) = \frac{M^K e^{-M}}{K!}$$

is taken to be Poisson distributed with M being constant. The number of non-zero components greatly affects the performance of sparse recovery algorithms and also depends on the environment. So in effect K models the number of multipath

components. One can easily imagine that K can also tend to ∞ in reality with the corresponding coefficients having near-zero values.

To model an exponentially-decaying profile, the coefficients are conditioned on their corresponding delays as a Gaussian distribution,

$$c_k|t_k \sim \mathcal{N}(0, \sigma^2(t_k)), \qquad \sigma^2(t_k) = ue^{-t_k/v}. \tag{8.2}$$

The model further contains two deterministic parameters; v which is the decay rate and u which is a normalisation constant. These two constants are considered to be given or somehow estimated since they represent the physical properties of the channel. They can be derived from an empirical model or expert knowledge about the channel. An exponentially decaying power-delay profile portrays a number of physical channels both over-the-air and underwater [48, 58]. It is possible that the power-delay profile of a channel to follow some different trend and not an exponential one. The model presented here is general enough to demonstrate the proposed methods but not limiting as to sacrifice generality.

In Figure 8.1 a simulated channel drawn from this model is shown. The sampling frequency was set to 10 MHz and $M = 8$. The continuous line represents the exponential trend used to draw the channel coefficients drawn as stems. The exponential decay rate was set to $v = 0.4\mu s$. The channel duration was set to $3.2\mu s$. On the right hand side of Figure 8.1 one can observe the troublesome nature of such a channel.

In Figure 8.2 the expected value $\langle c_k \rangle$ of 226 real-world channel responses is plotted [25]. The sampling rate for this specific experiment was $7.56MHz$ and the DTMB-A system under test was able to output the estimated channel at 256 taps. The lines above and below represent $\langle c_k \rangle \pm \sigma_{c_k}$, i.e, the mean value plus/minus one standard deviation. It is evident that the variance of the channel coefficients exhibits an exponential decay and though not strictly sparse, the channel can be considered as such since the number of significant components is small and placed at smaller delays.

8.2 OFDM Signal Model

Consider a perfectly synchronised, single-user OFDM system with N sub-carriers and a cyclic prefix (CP) of D samples, duration $T_{cp} = DT_s$ and sampling period T_s. The channel has a finite impulse response $\boldsymbol{c} = [c_1, \cdots, c_L]$ with $L \leq D$ and is stationary during the transmission of a single OFDM symbol.

The N-point inverse fast Fourier Transform (FFT) of a data block $\boldsymbol{s} = [s_1, \cdots, s_N]^T$

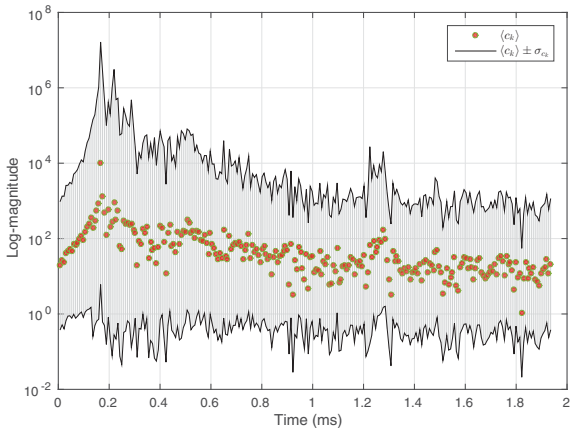


Figure 8.2: *Mean and variance of measured channel components.*

is computed and the CP is added to form the transmit OFDM symbol,

$$\boldsymbol{x}_{cp} = [x_{N-D+1}, \cdots, x_N, x_1, \cdots, x_N]^T$$

with $\boldsymbol{x} = \boldsymbol{F}^H \boldsymbol{s}$ and $\boldsymbol{F}$ the FFT matrix. The data in $\boldsymbol{s}$ usually come from a modulation scheme, e.g., QAM and have unit power. The received block of data is denoted as $\boldsymbol{r}_{cp} \in \mathbb{C}^{(N+D)\times 1}$.

$$\boldsymbol{r}_{cp} = \begin{bmatrix} r_1 \\ \vdots \\ r_D \\ r_{D+1} \\ \vdots \\ r_{N+D} \end{bmatrix}_{(N+D)\times 1} = \begin{bmatrix} \boldsymbol{C}_{ISI} & \boldsymbol{C}_{IBI} \end{bmatrix}_{(N+D)\times 2(N+D)} \begin{bmatrix} \boldsymbol{x}_{cp} \\ \boldsymbol{x}_{IBI} \end{bmatrix}_{2(N+D)\times 1}.$$

At the receiver the CP is discarded leading to the disappearance of the inter-block interference (IBI) imposed by matrix $\boldsymbol{C}_{IBI}$. The inter-block interference is a result of the previously transmitted symbol $\boldsymbol{x}_{IBI}$ still present on the channel during the transmission of $\boldsymbol{x}_{cp}$. Next it is easy to verify that

$$\begin{bmatrix} r_{D+1} \\ \vdots \\ r_{N+D} \end{bmatrix}_{N\times 1} = \underbrace{\begin{bmatrix} & c_L & \cdots & c_1 & \cdots & 0 \\ \boldsymbol{0}_{N\times(D-L+1)} & \vdots & \ddots & \ddots & c_1 & 0 \\ & 0 & \cdots & c_L & \cdots & c_1 \end{bmatrix}_{N\times(N+D)}}_{\text{lower part of } \boldsymbol{C}_{ISI}} \begin{bmatrix} x_{N-D+1} \\ \vdots \\ x_N \\ x_1 \\ \vdots \\ x_N \end{bmatrix}_{(N+D)\times 1}.$$

Based on the above and after an N-point FFT is taken, this can be conveniently

re-written as

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} r_{D+1} \\ \vdots \\ r_{N+D} \end{bmatrix} = \boldsymbol{F}\boldsymbol{C}\boldsymbol{F}^H\boldsymbol{s} + \boldsymbol{n} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{n} = \boldsymbol{S}\boldsymbol{h} + \boldsymbol{n}, \tag{8.3}$$

where $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}, \sigma_n^2\boldsymbol{I})$ denotes noise samples drawn from the Complex Normal distribution and $\boldsymbol{S} = \text{diag}(\boldsymbol{s})$ and superscript H denotes the Hermitian transpose. The second step is a direct consequence of the cyclic prefix being longer than the duration of the channel. Matrix $\boldsymbol{C}$ is the following circulant matrix,

$$\boldsymbol{C} = \begin{bmatrix} c_1 & 0 & \cdots & 0 & c_L & \cdots & c_3 & c_2 \\ c_2 & c_1 & \cdots & \cdots & 0 & c_L & \cdots & c_3 \\ \vdots & \ddots & \cdots & \ddots & \cdots & \ddots & \cdots & \vdots \\ c_L & \cdots & c_2 & c_1 & 0 & \cdots & \cdots & 0 \\ 0 & c_L & \cdots & c_2 & c_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \cdots & \ddots & \cdots & \ddots & \cdots & \vdots \\ 0 & \cdots & \cdots & c_L & c_{L-1} & \cdots & \cdots & c_1 \end{bmatrix}_{N \times N} = \boldsymbol{F}^H\boldsymbol{H}\boldsymbol{F}$$

where $\boldsymbol{H} = \text{diag}(\boldsymbol{h})$ and $\boldsymbol{h} = \boldsymbol{F}^{\boldsymbol{H}}\boldsymbol{c}$. This is due to the fact that every circulant matrix can be diagonalised by the Discrete Fourier Transform. Equalisation is then performed with an element-wise division of $\boldsymbol{y}$ with an estimate $\hat{\boldsymbol{h}}$ of the channel's frequency response.

8.2.1 Pilot-assisted Channel Estimation

Pilot symbols are transmitted at selected sub-carrier frequencies. Both the pilot symbols and their corresponding sub-carrier frequencies are known at the receiver. Let the set of pilot sub-carriers be $\mathcal{I} \subset [1, N]$ with $|\mathcal{I}| = m$. For the noiseless case it was shown in [96] that a Zero Forcing (ZF) equaliser can be realised as,

$$(\hat{\boldsymbol{h}}_{ZF})_{\mathcal{I}} = \boldsymbol{S}_{\mathcal{I}}^{-1}\boldsymbol{y}_{\mathcal{I}} \tag{8.4}$$

where the subscripted quantities indicate entries indexed by $\mathcal{I}$. The received symbols in $\boldsymbol{y}_{\mathcal{I}}$ are used for performing channel estimation and then equalisation of the rest of the symbols indexed by $[1, N] - \mathcal{I}$.

Far better performance can be achieved by formulating a sparse recovery problem with the help of a discrete dictionary [79, 74, 8]. Multipath fading channels tend to have a response that can be taken to be sparse. The experimental data shown in Figure 8.1 shows that only a few components closer to the receiver can contribute

significantly. In addition the channel samples is usual to have a decay trend as the time delay increases in the channel duration interval.

Assuming that $s_k = 1 + j0$ for $k \in \mathcal{I}$ then $\boldsymbol{y}_{\mathcal{I}}$ are samples of the frequency response,

$$\boldsymbol{S}_{\mathcal{I}}^{-1}\boldsymbol{y}_{\mathcal{I}} = \boldsymbol{h}_{\mathcal{I}} + \boldsymbol{S}_{\mathcal{I}}^{-1}\boldsymbol{n}$$

A partial Fourier matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times n}$ acts as as dictionary on a discrete set of delays $\mathcal{T} = \{i \cdot {}^{T_{cp}}/_{n} | 0 \leq i \leq T_{cp} - {}^{T_{cp}}/_{n}\}$,

$$\boldsymbol{\Phi}_{k,l} = e^{-2\pi\sqrt{-1}f_k\tau_l}, \;\; k \in \mathcal{I}, \; \tau_l \in \mathcal{T}. \tag{8.5}$$

where $f_k = {}^{k}/_{N \cdot T_s}$ is the k^{th} pilot sub-carrier frequency. A sparse recovery problem is formulated as follows,

$$\boldsymbol{S}_{\mathcal{I}}^{-1}\boldsymbol{y}_{\mathcal{I}} = \boldsymbol{\Phi}\hat{\boldsymbol{c}} + \boldsymbol{S}_{\mathcal{I}}^{-1}\boldsymbol{n} \tag{8.6}$$

where a suitable algorithm recovers a sparse vector $\hat{\boldsymbol{c}}$. An estimate $\hat{\boldsymbol{h}}$ is computed by extending the dictionary in Equation (8.5) to $k \in [1, N]$, i.e., all the sub-carrier frequencies.

8.3 Hierarchical Bayesian Channel Model

As discussed in Section 8.1, the multipath channel is taken to be sparse (or at least approximately) and its power-delay profile is exponentially decaying. One way to capture this prior statistical information is to use the following hierarchical Bayesian model. In particular, the complex channel coefficients c_i are modelled by the circularly-symmetric complex Gaussian random variable $c_i \sim \mathcal{CN}(0, \sigma_{c,i}^2)$. However, if the values of $\sigma_{c,i}^2$ are chosen as fixed constants, then the support set of the channel coefficients is fixed, that is, $c_i = 0$ with probability equal to one if $\sigma_{c,i}^2 = 0$ and $c_i \neq 0$ with probability equal to one otherwise. From what has been discussed, the sparsity pattern of the channel is decided by the wireless environment and subjects to temporal changes. This physical phenomenon can be described by the hierarchical Bayesian model which models the inverse variance $\sigma_{c,i}^{-2}$ of each sparse coefficient as a random variable. The certain choice of a Gamma hyper-prior is convenient because as it was discussed in Chapter 4, it is conjugate to $\mathcal{CN}(0, \sigma_{c,i}^2)$ thus making computations analytically tractable.

Let $\alpha_i^{-1} := \sigma_{c,i}^2$ and α_i be Gamma distributed

$$p(\alpha_i) = \text{Gamma}(a_i, b_i) = \frac{b_i^{a_i}}{\Gamma(a_i)}\alpha_i^{a_i - 1}e^{-b_i\alpha_i}$$

where $\Gamma(a_i)$ is the Gamma function. Then

$$p\left(\boldsymbol{c}|\boldsymbol{\alpha}\right) = \prod_{i=1}^{n} \mathcal{CN}\left(0, \alpha_i^{-1}\right) = \mathcal{CN}\left(\boldsymbol{0}, \boldsymbol{A}^{-1}\right)$$

where matrix $\boldsymbol{A} = \text{diag}([\alpha_1, \cdots, \alpha_n])$. The marginal of c_i is given by integrating out α_i from $p(c_i^{re}|\alpha_i)$ and $p(c_i^{im}|\alpha_i)$,

$$p(c_i^{re}) = p(c_i^{im}) = St(a_i, b_i) \tag{8.7}$$

where $c_i^{re} = \Re\{c_i\}$, $c_i^{im} = \Im\{c_i\}$ and St denotes the *Student-t* distribution with zero mean. Due to the heavy tails of the Student-t distribution, the channel coefficients c_i will be sparse. By designing the parameters a_i and b_i (see more details in Section 8.4), a desired power-delay profile can be achieved. Given the channel coefficients and the channel noise variance, the distribution of the received signal is given by

$$p(\boldsymbol{y}_{\mathcal{I}}|\boldsymbol{c}, \sigma_n^2) = \mathcal{CN}(\boldsymbol{\Phi c}, \sigma_n^2 \boldsymbol{I}).$$

Here, the noise variance σ_n^2 is known at the receiver, which is the common assumption in practice. Unlike traditional sparse recovery algorithms, this model allows the automatic determination of the number K of the multipath components which is a random variable.

The hyper-prior is a nodal point for this work as it allows to *express preference* for certain values of the variance of the multipath components. Note that given hyper-parameters α_i, the channel estimation problem reduces to the classic MMSE estimation for Gaussian random variables and admits a linear solution given by

$$\hat{\boldsymbol{c}} = \sigma_n^2 \left(\sigma_n^2 \boldsymbol{\Phi}^H \boldsymbol{\Phi} + \boldsymbol{A}\right)^{-1} \boldsymbol{\Phi}^H \boldsymbol{y}_{\mathcal{I}}.$$

To estimate the optimal values for α_i becomes the major task. This task can be achieved by using variational methods. The details of the iterative updates are given as follows. From the previous chapter; a factorised approximation to the posterior distribution is assumed,

$$p(\hat{\boldsymbol{c}}, \boldsymbol{\alpha}|\boldsymbol{y}_{\mathcal{I}}) \approx q(\hat{\boldsymbol{c}}) q(\boldsymbol{\alpha}),$$

where

$$q(\boldsymbol{\alpha}) = \prod_{i=1}^{n} \text{Gamma}(\tilde{a}_i, \tilde{b}_i)$$
$$q(\hat{\boldsymbol{c}}) = \mathcal{CN}(\boldsymbol{w}, \boldsymbol{\Sigma}).$$

The Variational Lower Bound (VLB) $\mathcal{L}$ is then maximised with respect to each of

the model parameters,

$$\mathcal{L} = \langle \ln p(\boldsymbol{y}_{\mathcal{I}}|\hat{\boldsymbol{c}}) \rangle + \langle \ln p(\hat{\boldsymbol{c}}|\boldsymbol{\alpha}) \rangle + \langle \ln p(\boldsymbol{\alpha}) \rangle - \langle \ln q(\hat{\boldsymbol{c}}) \rangle - \langle \ln q(\boldsymbol{\alpha}) \rangle$$

which gives the known update formulae as in Chapter 7:

$$\begin{aligned} &\boldsymbol{\Sigma} = \left(\sigma_n^2 \boldsymbol{\Phi}^H \boldsymbol{\Phi} + \boldsymbol{A}\right)^{-1}, \quad \boldsymbol{w} = \sigma_n^2 \boldsymbol{\Sigma} \boldsymbol{\Phi}^H \boldsymbol{y}_{\mathcal{I}} \\ &\tilde{a}_i = a_i + 1, \quad \tilde{b}_i = b_i + |w_i|^2 + \boldsymbol{\Sigma}_{i,i}, \quad \alpha_i = \tilde{a}_i / \tilde{b}_i. \end{aligned} \tag{8.8}$$

The avid reader will notice that there is a slight difference with the corresponding equations in Chapter 7 because of the use of the Complex Normal distribution.

8.3.1 Related Approaches

The Sparse Bayesian Learning framework has attracted the attention of researchers for the purpose of channel estimation with some of the contributions being distantly related to the purpose of this chapter. The most alluring aspects of this Hierarchical model for channel estimation is firstly its flexible nature to accommodate a plethora of situations and secondly the fact that the inference algorithms have a very nice mechanism of selecting the number of dominant components automatically without having to cope with fixed sparsity levels.

The authors in [47] have adopted the SBL model for the equalisation of a single input-output Pulse Amplitude Modulation (PAM) system with two levels, i.e., 2-PAM. The authors exploit the form of the cost function of a constant modulus algorithm to incorporate it into the SBL model. This approach falls well outside of the scope of this chapter since it aims at completely different modulation schemes.

The authors in [23] consider equalisation in OFDM systems by directly applying the FMLM algorithm (see Chapter 5) but by considering a slightly different implementation to allow tracking of the channel coefficients in a block-spaced manner. Basically the algorithm exploits the properties of the model to update the channel estimates by transmitting one pilot at a time. This leads to reduced pilot tone overhead.

The authors in [76] have proposed an inspired way to use the SBL framework to perform channel equalisation in OFDM systems. Basically they propose to use the Expectation-Maximisation approach to infer both the channel *and* the transmitted data. This done so as to achieve better performance in equalisation by using the full block of received symbols $\boldsymbol{y}$ and not just the received symbols related to the pilots $\boldsymbol{y}_{\mathcal{I}}$. Not that the authors assume a discrete channel and for this they assume $p(c_i|\alpha_i)$ for each channel tap and do not assume a dictionary for discretisation of the channel response.

From Chapter 5, in the traditional EM algorithm one computes the E-step (the posterior):

$$p(\boldsymbol{c}|\boldsymbol{y}_{\mathcal{I}}, \boldsymbol{\alpha}) = E_{\boldsymbol{c}|\boldsymbol{y}_{\mathcal{I}},\boldsymbol{\alpha}}[\log p(\boldsymbol{c}, \boldsymbol{y}_{\mathcal{I}}, \boldsymbol{\alpha})]$$

while at the M-step one maximises,

$$\hat{\alpha}_i = \arg\max_{\alpha_i>0} p(\boldsymbol{c}|\boldsymbol{y}_{\mathcal{I}}, \boldsymbol{\alpha}).$$

In their approach the authors argue that estimating $\boldsymbol{c}$ by using all of the available data in the block $\boldsymbol{y}$ will result in better performance while at the same time this will allow for estimating the data symbols as well in $\boldsymbol{s}_{[1,N]-\mathcal{I}}$. For this reason they consider at the E-step,

$$p(\boldsymbol{c}|\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\alpha}) = E_{\boldsymbol{c}|\boldsymbol{y},\boldsymbol{s},\boldsymbol{\alpha}}[\log p(\boldsymbol{c}, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\alpha})]$$

and at the M-step the joint maximisation of,

$$\{\hat{\alpha}_i, \hat{\boldsymbol{s}}\} = \arg\max_{\alpha_i>0,\boldsymbol{s}} p(\boldsymbol{c}|\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\alpha}).$$

For the M-step basically one then needs to perform two separate maximisation problems

$$\begin{aligned}\hat{\alpha}_i &= \arg\max_{\alpha_i>0} E_{\boldsymbol{c}|\boldsymbol{y},\boldsymbol{s},\boldsymbol{\alpha}}[\log p(\boldsymbol{x}, \boldsymbol{\alpha})] \\ \hat{\boldsymbol{s}} &= \arg\max_{\boldsymbol{s}} E_{\boldsymbol{c}|\boldsymbol{y},\boldsymbol{s},\boldsymbol{\alpha}}[\log p(\boldsymbol{y}|\boldsymbol{c}, \boldsymbol{s})].\end{aligned}$$

This is the basic idea behind their approach on how the SBL van be beneficial for channel equalisation.

The work in [74] adopts the continuous channel model described in Section 8.1 and the formulation of a sparse recovery problem like in Subsection 8.2.1 for pilot-assisted channel estimation in OFDM systems. The authors consider the model in Section 8.3 which they term as a 3-layer hierarchical model. A variational algorithm is developed for inference which is based on message passing and bears some resemblance with the traditional variational approach. The model makes no assumption on the type prior even though an uninformative prior is used as per the norm. The approach makes no attempt for improving the complexity of the algorithm.

8.4 Model Design from Real Data

Consider now the channel profile of Subsection 8.1 where the variance of $\hat{c}_i$ at $t_i \in \mathcal{T}$ is $\sigma^2(t_i) = u' e^{-t_i/v'}$. Based on empirical knowledge of the channel [25], v' and

u' can be defined. Then for every $\hat{c}_i$ the values of a_i, b_i have to be tuned to reflect this prior preference.

By calculating the variance of each magnitude $|\hat{c}_i|$,

$$\begin{aligned}\mathbb{E}[|\hat{c}_i|^2] &= \mathbb{E}[\Re(\hat{c}_i)^2] + \mathbb{E}[\Im(\hat{c}_i)^2] \\ &= 2\int \Re(\hat{c}_i)^2 p(\Re(\hat{c}_i)) d\Re\hat{c}_i = \frac{2b_i}{a_i - 1},\end{aligned}$$

which is a standard result for the Student-t distribution. In order to impose an exponential decay on the variance the following is imposed,

$$\frac{2b_i}{a_i - 1} = u' e^{-t_i/v'}. \tag{8.9}$$

Note that $E[\alpha_i] = {}^{a_i}/_{b_i}$ only depends on the ratio hence without loss of generality the following are defined,

$$\begin{aligned}a_i &= \frac{1}{2} u' e^{{}^{t_i}/_{v'}} + 1 \\ b_i &= 1.\end{aligned}$$

8.5 Extended Variational SBL

Recall from Chapter 7 that for a given index i, the values of α_j are fixed for all $j \neq i$, and only the update of α_i is considered. After some mathematical manipulation one arrives at an implicit expression for $\tilde{\alpha}_i^{r+1}$ at iteration $r+1$ as a function of $\tilde{\alpha}_i^r$,

$$\begin{aligned}\frac{(a_i + 1)}{\tilde{\alpha}_i^{[r+1]}} &= b_i + w_i^2 + z_i - \frac{z_i^2 + 2w_i^2 z_i}{{}^1/\tilde{\alpha}_i^{[r]} + z_i} + \frac{z_i^2 w_i^2}{({}^1/\tilde{\alpha}_i^{[r]} + z_i)^2} \\ &=: G(\tilde{\alpha}_i^{[r]}).\end{aligned} \tag{8.10}$$

where $z_i = \boldsymbol{e}_i^T \boldsymbol{\Sigma}_{-i} \boldsymbol{e}_i$, $w_i^2 = \sigma^2 \boldsymbol{e}_i^T \boldsymbol{\Sigma}_{-i} \boldsymbol{\Phi}^H \boldsymbol{y}_{\mathcal{I}} \boldsymbol{y}_{\mathcal{I}}^H \boldsymbol{\Phi} \boldsymbol{\Sigma}_{-i} \boldsymbol{e}_i$, $\boldsymbol{\Sigma}_{-i} = (\sigma^2 \boldsymbol{\Phi}^H \boldsymbol{\Phi} + \tilde{\boldsymbol{A}}_{-i})^{-1}$, G is the so-called map function and $\boldsymbol{e}_i$ is the i^{th} canonical vector. Notation $\boldsymbol{\Phi}_{-i}$ means the removal of column i while $\tilde{\boldsymbol{A}}_{-i}$ of both the row and column. Please note that the equation above is slightly different from the one in the previous chapter.

Equation (8.10) specifies the update rule for α_i from one iteration to the next. Letting the iteration $r \to \infty$ one has $\tilde{\alpha}_i^{[r+1]} = \tilde{\alpha}_i^{[r]} = \tilde{\alpha}_i^{\infty}$. It has been shown that for the case of $a_i = b_i = 0$, by solving

$$\tilde{\alpha}_i^{\infty} - G\left(\tilde{\alpha}_i^{\infty}\right) = 0, \tag{8.11}$$

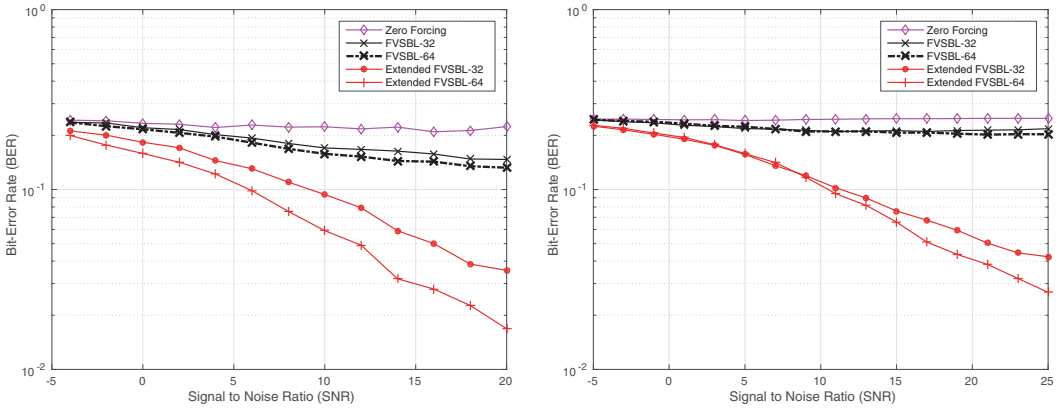


Figure 8.3: *(a) Empirical BER performance for simulated channels (b) BER for real-world channel responses.*

two asymptotically stable fixed points are found:

$$\tilde{\alpha}_i = \begin{cases} (w_i^2 - z_i)^{-1}, & w_i^2 > z_i \\ +\infty & w_i^2 \leq z_i. \end{cases} \tag{8.12}$$

If the variance of a multipath $\alpha_i = +\infty$, then $\hat{c}_i$ is found not to be active. In the case of an *informative prior* ($a_i \neq 0, b_i \neq 0$) the fixed point analysis is not able to be achieved and therefore the FVSBL cannot be applied to the problem considered here.

The Extended FVSBL is employed which can handle the informative prior efficiently via a very practical rule. Basically the *three* possible fixed points of Equation (8.11) are computed analytically. In the case of imaginary roots then the real one is selected. The following proposes a way out of the difficult case where three real fixed points are found.

Conjecture 2 ([52]). *If $w_i^2 > z_i$ and the solution of Equation* (8.11) *results in three distinct real roots $0 < \tilde{\alpha}_i^1 < \tilde{\alpha}_i^2 < \tilde{\alpha}_i^3 < +\infty$ then $\tilde{\alpha}_i^1$ causes the variational lower bound $\mathcal{L}$ to increase the most.*

Conjecture 2 recommends that the fixed point which achieves the greatest increase in the VLB is the one with the smallest value. This provides a complete selection rule for the fixed points even though an analytical expression like in (8.11) cannot be attained.

8.6 Test Cases

8.6.1 Synthetic OFDM System

A comparison is made between the *uninformative* FVSBL ($a_i = b_i = 0$), the Extended FVSBL ($a_i \neq 0, b_i \neq 0$) and a ZF equaliser with synthetic datasets. Convergence for the Extended FVSBL is reached once the change in the VLB between two iterations falls below 10^{-7}. Convergence of the uninformative FVSBL is tested like in [86] by checking whether the number of $\tilde{\alpha}_i$ that satisfy Equation (8.12) has remained the same *and* the ℓ_2 norm of the difference of the corresponding $\tilde{\alpha}_i$ between two iterations has fallen below 10^{-7}. Both algorithms are allowed to run for a maximum of 1000 iterations and are both initialised with empty models ($\tilde{\alpha}_i = +\infty$).

An OFDM system with 512 sub-carriers and $F_s = 10$ MHz is simulated. The guard interval is $T_{cp} = {}^{32}/_{F_s}$. Two cases of 32 and 64 pilot symbols are simulated. The modulation chosen was 4-QAM and the pilot symbols where set to $1 + j0$. A multipath channel was simulated with Poisson parameter $K = 10$ and additive white Gaussian noise. The decay rate was set to $v = {}^{4}/_{F_s}$ and u was set so that the components were normalised. The time delays of the multipath components where chosen from the interval $[0, {}^{16}/_{F_s}]$ uniformly at random. The dictionary was built with 64 atoms according to Equation 8.5. A prior was constructed like in Section 8.4; the decay rate was set to be $v' = v = {}^{10}/_{F_s}$ and $u' = u$. The values of a_i where set according to Equation (8.9) for the Extended FVSBL algorithm. This assumes exact knowledge of the decay rate.

In Figure 8.3a the average BER performance of 200 runs is plotted against a range of SNR values. By inspecting the FVSBL-32 and the Extended FVSBL-32 curves (32 pilot symbols) one notices the difference in performance by employing the prior. Both perform better than the ZF because of the sparsity assumption. The informative algorithm achieves the lowest BER curve even for low SNR values. Looking at the curves for 64 pilot symbols, the performance for the informative case is still far better than the uninformative case which has shown only marginal improvement over the case with 32 pilot symbols. This is an important aspect since the pilot symbol overhead can be reduced given appropriate CSI.

In Figure 8.4 the runtime is compared. The Extended FVSBL converges faster than its uninformative counterpart especially when the problem size increases. The curves suggest that for 64 pilot symbols the informative algorithm performs as fast as the uninformative for half the number of symbols. The proposed algorithm handles the statistical bias cleverly achieving faster convergence *and* better performance.

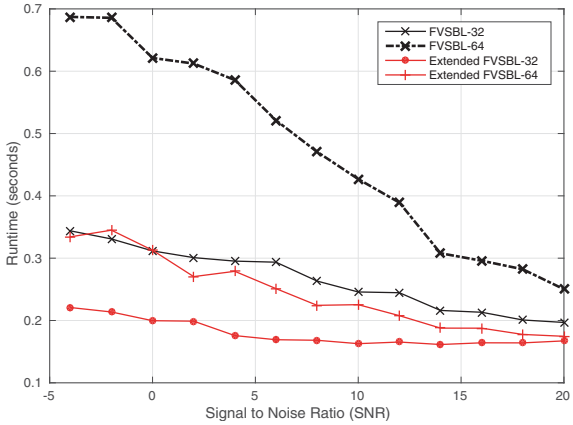


Figure 8.4: *Runtime performance (ZF not shown).*

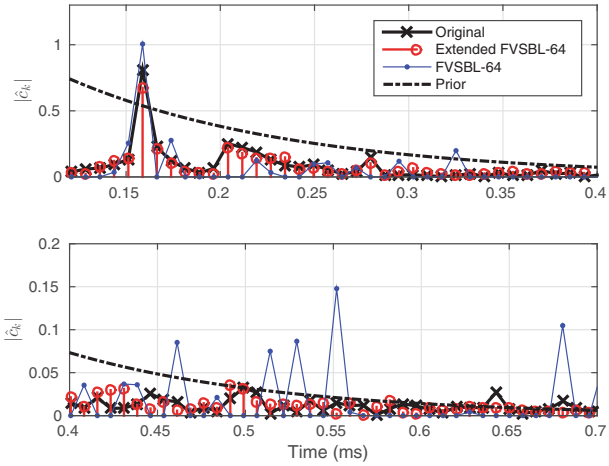


Figure 8.5: *Comparison of reconstructed responses and prior.*

8.6.2 Real-world OFDM System

Here the dataset mentioned in Subsection 8.1 is used, which consists of 226 channel responses. The first 113 channel responses are used to compute the sample variance for each of the 256 taps. Then a prior is empirically constructed $v = {}^{16}/_{F_s}$. This can be seen in Figure 8.5 (dashed line) which follows the trend of one of the remaining responses (crossed line).

An OFDM system like before is simulated with 1024 sub-carriers and 4-QAM. The guard interval was set to be $T_{cp} = {}^{256}/_{F_s}$, i.e., the length of the responses. The dictionary was also constructed in the same manner. Two cases of 32 and 64 pilot symbols were considered.

In Figure 8.3b the average BER performance over 200 runs is shown. At each run a response was chosen randomly out of the remaining 113 that *did not* participate in constructing the prior. It is pointed out that the Extended FVSBL algorithm performs far better than its uninformative cousin in both cases of pilot symbols.

Back to Figure 8.5, the reconstructed channel responses are plotted against one

of the actual ones for the case of SNR=25db and 64 pilots. In the top half, focus is turned on the smaller delays where the Extended FVSBL algorithm has managed to capture the trend *and* the magnitude of almost all of the components while the uninformative has produced large components where they do not exist. The same is observed in the bottom half where the Extended FVSBL imposes the trend for small components based on the prior.

8.7 Conclusion

This chapter has focused on a very fitting application of the Extended FVSBL algorithm for multipath channel equalisation. A common attribute of sparse multipath channels was exploited, i.e., the fact that the channel coefficients are not only sparse but they follow an exponential decay law as well. A rather simple profile was adopted for the channel coefficients but in reality this sort of information may originate from an empirical model or expertise. This information finds a very nice way into the sparse Bayesian learning model as the tuning hyper-parameters of the hyper-prior distribution over the channel coefficients. In this way it is possible to have a fully flexible channel model able to absorb any power-delay profile by an appropriate tuning of the hyper-prior. For this task the variational approach to SBL is the most fitting as was shown in Chapter 7. The Extended FVSBL is used for inferring the channel parameters from the OFDM training symbols after casting the equalisation process as a sparse recovery problem. The prior distribution assumed for the channel coefficients is informative which precludes the traditional fast variational approach of being used.

Chapter 9

Approximate Message Passing

A different class of sparse recovery algorithms based on Bayesian methods is that of Message Passing. They have been used for sparse recovery in various examples with the most widespread that of [6] with subsequent modifications like that of [88]. The message passing is based on a certain type of representation of a probabilistic model and the derivation of an algorithm for performing inference over such a graph by viewing the relation ship between the nodes of the graph as exchanged messages. In general such algorithms succeed in providing the exact formula for marginal distributions for many types of models. In the case of sparsity-promoting models like the ones that have been introduced earlier these message passing algorithms provide approximations.

Message passing has been the initial step for a series of "approximations" and assumptions with the most important one being that of the very-large system. Probably the motivation behind such approximations to an already approximate algorithm was their overly expensive computational nature. Fortunately these approximations that will be discussed next have provided a very powerful class of algorithms termed Approximate Message Passing that exhibit *extremely* low complexity and some quite favourable properties for a new type of analysis; the State Evolution formalism.

Even though the Sparse Bayesian Learning models have been widely accepted by the engineering community and for a countless applications it is still uncertain what the relationship is between the Type-II approximate inference algorithm and that suggested by Approximate Message Passing. The main motivation to explore such relationships is basically the fact that both are built on a hierarchical model of sparsity-inducing prior distributions. Also the fact that if such relationships become known then further improvements could be made possible by combining the two approaches.

9.1 Inference on Graphical Models

In Chapter 4 the notion of Bayesian networks was introduced and it was evident immediately that one can communicate and expand a probabilistic model with ease and intuition. All the Bayesian networks that have been presented so far had one thing in common; the fact that the joint distribution was able to be separated into individual factors. The Bayesian network of course hides this information and this is where the factor graphs come into play.

In the hierarchical Bayesian networks for Sparse Bayesian Learning, inference was performed approximately via the Type-II maximum likelihood procedure which resulted in a certain class of inference algorithms. The idea behind factor graphs is to devise inference algorithms that are directly related to the graph itself.

9.1.1 Factor Graphs

The example is borrowed from [9]. Consider the following joint distribution of random variables x_1, x_2, x_3,

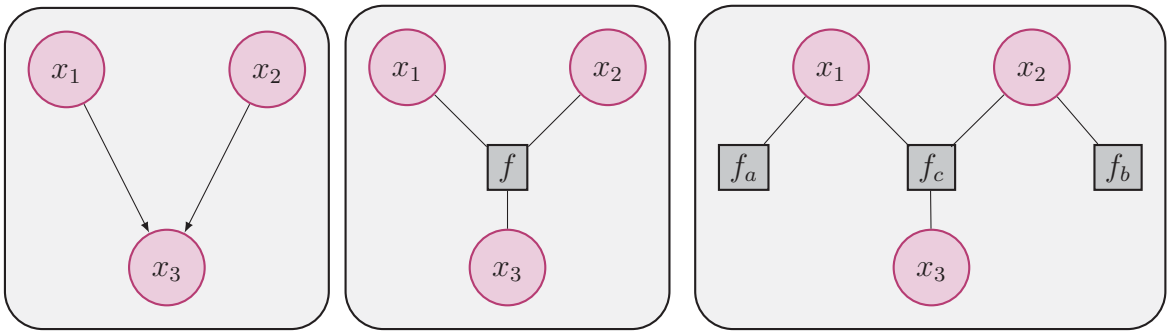
$$f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2).$$

In Figure 9.1a the corresponding Bayesian network shows the dependency between the random variables. In Figures 9.1b and 9.1c two renditions for the corresponding factor graph are shown. One quickly notices that the factor graphs include only two types of nodes, the so called *variable nodes* and the *factor nodes* depicted by circles and squares respectively. As one might guess the factor nodes correspond to the factors that make up the joint distribution related to the graph. So for Figure 9.1b the single factor node is actually the same as the joint distribution. For Figure 9.1c the factor nodes are,

$$f_a(x_1) = p(x_1), \quad f_b(x_2) = p(x_2), \quad f_c(x_1, x_2, x_3) = p(x_3|x_2, x_1).$$

Each factor node is associated with all the neighbouring variables and vice-versa.

Comparing the notion between Bayesian networks and Factor graphs it is evident that factor graphs bear no information about the underlying distributions but focus solely on the hierarchical structure of the model. Factor graphs can represent the form of the factorised joint distribution precisely. By using factor graphs it is possible to derive algorithms for performing inference by the means of the messages sent from one node to another. The notion of messages refers to the distribution functions computed at the connecting edges between the nodes. The messages then can be used to compute marginal distributions for any part of the graph.



(a) *Bayesian Network for* $f(x_1, x_2, x_3)$.

(b) *The corresponding factor graph.*

(c) *Another rendition of a factor graph for the same joint distribution.*

Figure 9.1: *Example showing the transition of a Bayesian network to a factor graph.*

9.1.2 The Sum-Product Algorithm

For notational convenience, letters a, b will be representing the index variables for *factor nodes* while letters i, j will be representing index variables for *variable nodes*. This is done to simplify notation as much as possible. For this purpose the messages from variable nodes x_i to factor nodes f_a will be represented as

$$\mu_{i\to a}(x_i)$$

while messages from factor nodes to variable nodes will be given by

$$\mu_{a\to i}(x_i).$$

The sum-product algorithm is an efficient way of performing exact inference on factor graphs given the fact that there are no cycles in the structure of the graph. This assumption can be violated but the resulting algorithm will be approximate. This is known as the "*loopy*" version of the sum-product algorithm and has been used with great success. Recall that in earlier chapters approximate inference was also used in the case of Type-II Maximum likelihood. In both cases the main fact was that the joint distribution of all the variables in the model can be written as a product of factors.

The actions of the algorithm can be divided into two categories, messages sent from variable nodes to factor nodes as shown in Figure 9.2a and as messages sent from factor nodes to variable nodes shown in Figure 9.2b. Focusing only on the main results (a full analysis is located in [9]), the outgoing message is

$$\mu_{i\to a}(x_i) = \prod_{b\neq a} \mu_{b\to i}(x_i),$$

which in writing means that the variable node sends out the product of all the incoming messages except the message coming from the factor node it is sending the message to.

Respectively for the outgoing messages from factor nodes,

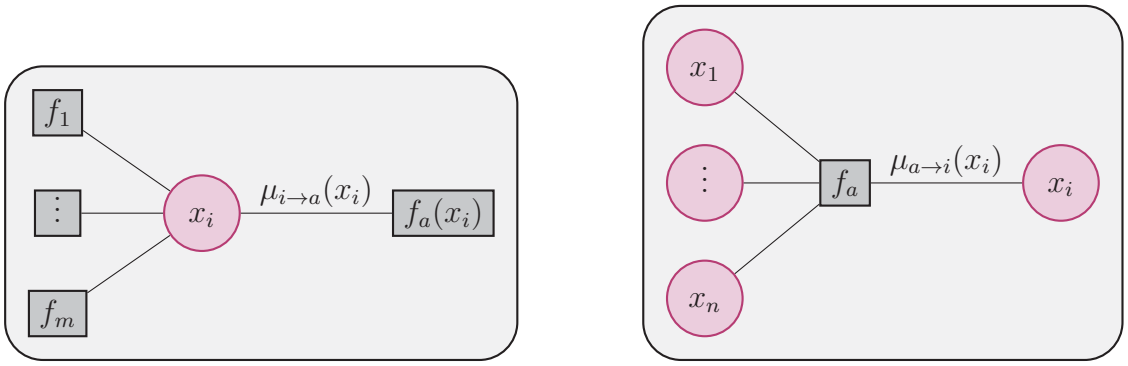
$$\mu_{a\to i}(x_i) = \int f_a(x_i, x_1, \cdots, x_n) \prod_{j\neq i} \mu_{j\to a}(x_i) dx_{j\neq i}$$

which means to multiply the function associated with the factor node with all the other incoming messages (accept the one coming from x_i) and then integrate all over those variables.

The simple cases where a variable or a factor node is a *leaf*, i.e., the node has only one associated edge the messages are then simply

$$\mu_{i\to a}(x_i) = 1$$
$$\mu_{a\to i}(x_i) = f_a(x_i).$$

Usually these types of messages are absorbed into their neighbouring nodes.



(a) *From variable node x_i to factor node $f_a(x_i)$.*

(b) *From factor node $f_a(x_1, \cdots, x_n)$ to variable node x_i.*

Figure 9.2: *Types of messages exchanged between nodes of a factor graph.*

Once all the messages have been propagated then the marginal distribution for any variable x_i can be computed by forming the product of all the incoming messages to that variable node (from neighbouring factor nodes),

$$p(x_i) \propto \prod_a \mu_{a\to i}(x_i).$$

Likewise, the marginal over the variables associated with a factor node $f_a(x_1, \cdots, x_n)$ can be computed by multiplying all the incoming messages (from neighbouring vari-

able nodes) with the function related to that node,

$$p(x_1, \cdots, x_n) = f_a(x_1, \cdots, x_n) \prod_i \mu_{i \to a}(x_i).$$

9.2 Derivation of the AMP Algorithm

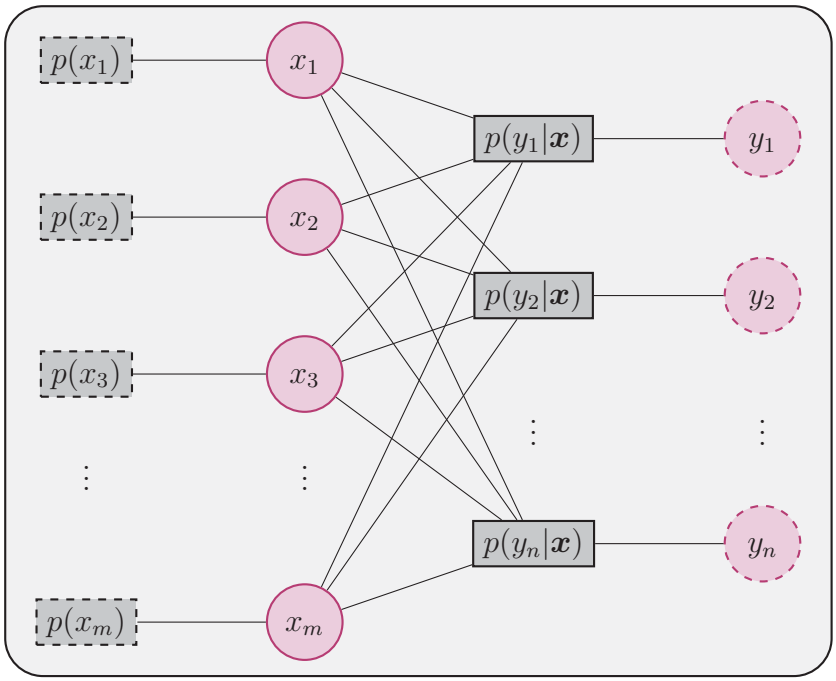


Figure 9.3: *The factor graph corresponding to the joint distribution used in the AMP algorithm.*

The Factor graph in Figure 9.3 corresponds to the following joint distribution,

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{x}) &= p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}) \\ &= \prod_{a=1}^{n} p(y_a|\boldsymbol{x}) \prod_{i=1}^{m} p(x_i) \end{aligned}$$

where the likelihood function as usually is given by

$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{H}\boldsymbol{x}, \sigma^2 \boldsymbol{I}).$$

The prior distribution placed independently over each x_i is a Laplace prior,

$$p(x_i) = \frac{\lambda}{2\sigma^2} e^{-\lambda/\sigma^2 |x_i|}.$$

It was discussed in Chapter 4 how the Laplace distribution leads to sparse estimates. The AMP algorithm builds heavily on this model even though extensions to accommodate arbitrary distributions have been proposed [81].

To further reduce clutter the factor and variable nodes that are drawn with dashed lines in Figure 9.3 are not taken into consideration since they represent the prior distribution and the deterministic measurements are both given as input and can be absorbed in their neighbours.

Based on the main results for the sum-product algorithm discussed earlier, the posterior distribution for any factor node is,

$$p(x_i|\boldsymbol{y}) = p(x_i) \prod_{a\in[1,n]} \mu_{a\to i}(x_i).$$

9.2.1 Applying the Sum-Product Algorithm

Applying the direct result from the sum-product algorithm for messages from factor nodes to variable nodes we obtain,

$$\mu_{a\to i}(x_i) = \int p(y_a|\boldsymbol{x}) \prod_{j\neq i} \mu_{i\to a}(x_i) dx_{j\neq i} \propto \int e^{-\frac{1}{2\sigma^2}(y_a-(\boldsymbol{H}\boldsymbol{x})_a)^2} \prod_{j\neq i} \mu_{j\to a}(x_i) dx_{j\neq i} \tag{9.1}$$

Applying the result from the sum-product algorithm for messages from variable nodes to factor nodes, it is straightforward to write,

$$\mu_{i\to a}(x_i) \propto e^{-\frac{\lambda|x_i|}{\sigma^2}} \prod_{b\neq a} \mu_{b\to i}(x_i). \tag{9.2}$$

By inspecting Equations (9.1) and (9.2) it is easy to verify that indeed there has to be a message passing schedule since for the computation of either of the equations the other one is needed. This is in agreement with the sum-product theory which dictates that inference is exact for *acyclic* graphs which is not the case with the factor graph in Figure 9.3.

The algorithm is transformed then in an iterative scheme for passing messages based on the iteration t,

$$\mu^{t+1}_{i\to a}(x_i) \propto e^{-\frac{\lambda|x_i|}{\sigma^2}} \prod_{b\neq a} \mu^t_{b\to i}(x_i)$$

$$\mu^t_{a\to i}(x_i) \propto \int e^{-\frac{1}{2\sigma^2}(y_a-(\boldsymbol{H}\boldsymbol{x})_a)^2} \prod_{j\neq i} \mu^t_{j\to a}(x_j) dx_{j\neq i} = \mathrm{E}_{x_{j\neq i}}\left[e^{-\frac{1}{2\sigma^2}(y_a-(\boldsymbol{H}\boldsymbol{x})_a)^2}\right].$$

In the last part it was identified that the message is in fact the expectation of $e^{-\frac{1}{2\sigma^2}(y_a-(\boldsymbol{H}\boldsymbol{x})_a)^2}$ for the random variables $x_{j\neq i}$ and their respective distributions given by the messages $\mu_{j\to a}(x_j)$.

9.2.2 The large system limit

One of the basic assumptions is that of the large system limit, i.e, when the dimensions of the problem $n, m \to \infty$ and $\delta \to \frac{n}{m}$. In this limit it is shown that the afore mentioned messages can in fact be approximated by the means well-known distributions. This is tied closely to the fact that the entries of matrix $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ are taken to be independent and identically distributed random variables drawn from $\mathcal{N}(0, \frac{1}{n})$. Furthermore the columns of matrix $\boldsymbol{H}$ are normalised to unit ℓ_2-norm. Hence in the large system limit the following hold

$$\sum_{a=1}^{n} H_{ai} = 0, \quad \sum_{a=1}^{n} H_{ai}^2 = 1$$
$$\sum_{b\neq a} H_{bi}^2 \approx 1 - \frac{1}{n}, \quad H_{ai} = O(1/\sqrt{n}).$$

Approximating $\mu_{a\to i}(x_i)$

For this take the following convenient rewriting,

$$\mu_{a\to i}(x_i) \propto \mathrm{E}_{x_{j\neq i}}\left[e^{-\frac{1}{2\sigma^2}\left(y_a - H_{ai}x_i - \sum_{j\neq i} H_{aj}x_j\right)^2}\right] = \mathrm{E}_{x_{j\neq i}}\left[e^{-\frac{1}{2\sigma^2}(z - H_{ai}x_i)^2}\right]$$

where the auxiliary random variable

$$z = y_a - \sum_{j\neq i} H_{aj}x_j$$

was defined and will play a central role.

Furthermore the following two helpful definitions are made for a variable distributed according to $x_i \sim \mu_{i\to a}^t(x_i)$,

$$\mathrm{E}[x_i] = x_{i\to a}^t\ , \quad \mathrm{var}[x_i] = \sigma^2\tau_{i\to a}^t.$$

Then the mean and variance of z can be written down as

$$\mathrm{E}[z] = z_{a\to i}^t = y_a - \sum_{j\neq i} H_{aj}\mathrm{E}[x_j] = y_a - \sum_{j\neq i} H_{aj}x_{j\to a}^t \tag{9.3}$$
$$\mathrm{var}[z] = \hat{\tau}_{a\to i}^t = \sum_{j\neq i} H_{aj}^2\mathrm{var}[x_i] = \sigma^2\sum_{j\neq i} H_{aj}^2\tau_{j\to a}^t \approx \hat{\tau}^t.$$

In the last part the individual variances are assumed to be approximated by a single variance since the value of H_{aj}^2 is expected to be small for $n \to \infty$ thus dropping the dependence on index i.

A direct application of the Berry-Essen theorem [33, 1] shows that for $n \to \infty$,

$$\mu^t_{a\to i}(x_i) \propto \mathcal{N}\left(z^t_{a\to i}, \sigma^2(1+\hat{\tau}^t)\right) \tag{9.4}$$

which in effect means that the distribution conveyed by the message $\mu_{a\to i}(x_i)$ converges to a Gaussian, parametrised by the auxiliary random variable z. The main results from the application of the Berry-Esseen are not presented here since they are not entirely relevant to the main theme of this chapter.

Approximating $\mu_{i\to a}(x_i)$

Substituting the approximation for $\mu^t_{a\to i}(x_i)$ into the expression for $\mu^{t+1}_{i\to a}$

$$\begin{aligned}\mu^{t+1}_{i\to a}(x_i) &\propto e^{-\frac{\lambda|x_i|}{\sigma^2}} \prod_{b\neq a} \mu^t_{b\to i}(x_i) \\ &= e^{-\frac{\lambda|x_i|}{\sigma^2}} \prod_{b\neq a} \mathcal{N}\left(H_{ai}x_i | z^t_{b\to i}, \sigma^2(1+\hat{\tau}^t)\right).\end{aligned}$$

The exponent of the product of Gaussians in the expression above can be approximated as,

$$-\frac{1}{2\sigma^2(1+\hat{\tau}^t)} \sum_{b\neq a} \left(x_i^2 H_{bi}^2 + (z^t_{b\to i})^2 - 2x_i H_{bi} z^t_{b\to i}\right) \approx C - \frac{1}{2\sigma^2(1+\hat{\tau}^t)} \left(x_i - \sum_{b\neq a} H_{bi} z^t_{b\to i}\right)^2.$$

where constant $C = -\sum_{b\neq a}(z^t_{b\to i})^2$. The fact that $\sum_{b\neq a} H^2_{bi} \approx 1 - \frac{1}{n}$ was also used based on the properties of matrix $\boldsymbol{H}$.

Based on this approximation the message $\mu^{t+1}_{i\to a}(x_i)$ can be written as,

$$\mu^{t+1}_{i\to a}(x_i) \propto e^{-\frac{\lambda|x_i|}{\sigma^2} - \frac{1}{2\sigma^2(1+\hat{\tau}^t)}\left(x_i - \sum_{b\neq a} H_{bi} z^t_{b\to i}\right)^2}.$$

9.2.3 The noiseless case

In the limit of $\sigma^2 \to 0$ the variance of the auxiliary variable in Equation (9.3) becomes $\hat{\tau}^t = 0$. The value of $x^{t+1}_{i\to a}$ can be computed straightforwardly by using the Laplace's method for approximating integrals of the form $\int e^{M\cdot f(x)} dx$ when $M \to \infty$. Computing the mean,

$$\begin{aligned}x^{t+1}_{i\to a} = \mathrm{E}[x_i] &= \int x_i \mu^{t+1}_{i\to a}(x_i) dx_i \\ &\propto \int x_i e^{-\frac{\lambda|x_i|}{\sigma^2} - \frac{1}{2\sigma^2}\left(x_i - \sum_{b\neq a} H_{bi} z^t_{b\to i}\right)^2} dx_i\end{aligned}$$

one arrives at an integral which can be approximated by Laplace method as $\sigma^2 \to 0$. Laplace's method dictates that the value of the integral will be

$$x_{i\to a}^{t+1} \approx \arg\max_{x_i} \left\{ -\lambda|x_i| - \frac{1}{2}\left(x_i - \sum_{b\neq a} H_{bi} z_{b\to i}^t \right)^2 \right\}.$$

By computing the stationary points of the above expression it is concluded that,

$$x_{i\to a}^{t+1} = \begin{cases} \sum_{b\neq a} H_{bi} z_{b\to i}^t - \lambda, & \text{for} \quad \sum_{b\neq a} H_{bi} z_{b\to i}^t > \lambda \\ \sum_{b\neq a} H_{bi} z_{b\to i}^t + \lambda, & \text{for} \quad \sum_{b\neq a} H_{bi} z_{b\to i}^t < -\lambda \\ 0, \text{ otherwise .} \end{cases}$$

The above expression is better known as the *soft-threshold function* and can be written more compactly as

$$x_{i\to a}^{t+1} = \eta\left(\sum_{b\neq a} H_{bi} z_{b\to i}^t, \lambda \right), \tag{9.5}$$

which means that the threshold (second operand) decides the value of the outcome $x_{i\to a}^{t+1}$.

Now to compute the variance,

$$\begin{aligned} \tau_{i\to a}^{t+1} &= \frac{1}{\sigma^2}\text{var}[x_i] \\ &\propto \int (x_i - \text{E}[x_i])^2 \, e^{-\frac{\lambda|x_i|}{\sigma^2} - \frac{1}{2\sigma^2}\left(x_i - \sum_{b\neq a} H_{bi} z_{b\to i}^t\right)^2} dx_i \end{aligned}$$

Computing the expression above for $\sigma^2 \to 0$ results in a very simple interpretation given in [1] but it can also be derived rigorously based on the properties of exponential distributions,

$$\tau_{i\to a}^{t+1} = \begin{cases} 1, \text{ for } |\sum_{b\neq a} H_{bi} z_{b\to i}^t| \geq \lambda \\ 0, \text{ otherwise .} \end{cases}$$

This can be written as the derivative of the soft threshold function,

$$\tau_{i\to a}^{t+1} = \eta'\left(\sum_{b\neq a} H_{bi} z_{b\to i}^t, \lambda \right).$$

By dropping the dependence on the missing index a the following approximation

can be employed,

$$\tau^{t+1} = \frac{1}{n}\sum_{i=1}^{m} \tau_{i\to a}^{t+1} \tag{9.6}$$

where the variances are averaged over n factor nodes.

Summarising so far, in the large system limit and under the noiseless assumption; the message $\mu_{a\to i}^{t}$ becomes a deterministic value given by the mean of the Gaussian in Equation (9.4),

$$z_{a\to i}^{t} = y_a - \sum_{j\neq i} H_{aj} x_{j\to a}^{t}.$$

Likewise, the message $\mu_{i\to a}^{t+1}(x_i)$ can be approximated by a distribution with its mean given by Equation (9.5),

$$x_{i\to a}^{t+1} = \eta\left(\sum_{b\neq a} H_{bi} z_{b\to i}^{t}, \lambda\right).$$

9.2.4 First Order Approximation

In the two equations derived above for $z_{a\to i}^{t}$ and $x_{i\to a}^{t+1}$ the sums on the right hand side are the main problem to what otherwise would have been a very simple iteration, i.e., the sums for computing each of the messages are always one term short of the whole sum. This leads to making the following assumptions

$$\begin{aligned} x_{i\to a}^{t} &= x_i^{t} + \delta x_{i\to a}^{t} \\ z_{a\to i}^{t} &= z_i^{t} + \delta z_{a\to i}^{t}. \end{aligned} \tag{9.7}$$

So in reality it is assumed that the messages no longer depend on the missing indices a and i respectively, but that there is a correction term that does depend on them. The natural consequence of things is to approximate this correction term. In this approximation the properties of matrix H help greatly.

Substituting Equations (9.7) in the corresponding message variables and expanding the sum,

$$\begin{aligned} z_a^{t} + \delta z_{a\to i}^{t} &= y_a - \sum_{j} H_{aj}(x_j^{t} + \delta x_{j\to a}^{t}) + H_{ai} x_i^{t} + \cancelto{0}{H_{ai}\delta x_{i\to a}^{t}} \\ x_i^{t+1} + \delta x_{i\to a}^{t+1} &= \eta\left(\sum_{b} H_{bi}(z_b^{t} + \delta z_{b\to i}^{t}) - H_{ai} z_a^{t} + \cancelto{0}{H_{ai}\delta x_{a\to i}^{t}}, \lambda\right). \end{aligned} \tag{9.8}$$

Next the first order Taylor approximation of the second equation is taken at the

point $\sum_b H_{bi}(z_b^t + \delta z_{b\to i}^t)$,

$$x_i^{t+1} + \delta x_{a\to i}^{t+1} \approx \eta\left(\sum_b H_{bi}(z_b^t + \delta z_{b\to i}^t), \lambda\right) - \eta'\left(\sum_b H_{bi}(z_b^t + \delta z_{b\to i}^t), \lambda\right) H_{ai} z_a^t.$$

Then by inspection of the equations above one can identify the following

$$z_a^t = y_a - \sum_j H_{aj}(x_j^t + \delta x_{j\to a}^t) \tag{9.9}$$

$$\delta z_{a\to i}^t = H_{ai} x_i^t \tag{9.10}$$

$$x_i^{t+1} = \eta\left(\sum_b H_{bi}(z_b^t + \delta z_{b\to i}^t), \lambda\right) \tag{9.11}$$

$$\delta x_{i\to a}^{t+1} = -\eta'\left(\sum_b H_{bi}(z_b^t + \delta z_{b\to i}^t), \lambda\right) H_{ai} z_a^t. \tag{9.12}$$

It is now straight forward to eliminate the approximating terms by making the necessary substitutions. Plugging Equation (9.10) into Equation (9.11),

$$x_i^{t+1} = \eta\left(\sum_b H_{bi} z_b^t + \sum_b H_{bi}^2 x_i^t, \lambda\right)$$

which in vector form becomes

$$\boldsymbol{x}^{t+1} = \eta\left(\boldsymbol{x}^t + \boldsymbol{H}^T \boldsymbol{z}^t, \lambda\right)$$

with η being applied element-wise on its operands.

Plugging Equation (9.12) into Equation (9.9),

$$z_a^t = y_a - \sum_j H_{ai} x_j^t + \sum_j H_{aj}^2 \eta'(x_j^{t-1} + (\boldsymbol{H}^T \boldsymbol{z}^{t-1})_j, \lambda) z_a^{t-1}$$

$$\boldsymbol{z}^t \approx \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}^t + \frac{1}{n}\sum_j \eta'(x_j^{t-1}, \lambda) = \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}^t + \frac{1}{n}\|\boldsymbol{x}^t\|_0 \boldsymbol{z}^{t-1}$$

where in the final step the fact that the summation of the individual terms $\eta'(x_j^t, \lambda)$ actually returns the 0-norm of vector $\boldsymbol{x}^t$.

Finalising the discussion around the derivation of the AMP algorithm the itera-

tions are as follows,

$$\boxed{\begin{aligned} \boldsymbol{x}^{t+1} &= \eta\left(\boldsymbol{x}^t + \boldsymbol{H}^T\boldsymbol{z}^t, \theta_t\right) \\ \boldsymbol{z}^t &= \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}^t + \frac{1}{n}\|\boldsymbol{x}^t\|_0\boldsymbol{z}^{t-1} \\ \theta^t &= \frac{\alpha}{\sqrt{n}}\|\boldsymbol{z}^t\|_2. \end{aligned}} \tag{9.13}$$

The meaning and definition of the threshold θ^t will be made more clear in the discussion to follow.

To keep in line with the main bibliography around the AMP algorithm [69, 33, 32] the threshold from the Laplace prior λ is replaced with a scaled version based on the theory developed in [69],

$$\lambda = \theta_t(1 - b_t)$$

where $b_t = \frac{1}{n}\|\boldsymbol{x}^t\|_0$. This means that when the iterations above converge to a solution then this solution is the same as the solution given by the ℓ_1 approach with λ as the regularisation parameter.

9.3 State Evolution

One of the most prominent questions among theorists and engineers that deal with compressed sensing in one way or another is that of recovery and the conditions under which it will happen. From the early years of the appearance of Compressed Sensing, theorems relate the recovery conditions based on the sensing matrix, the sparsity of the signal and the noise level (a summary can be traced back in Chapter 2). Put simply and in simple engineering terms;"Will this matrix do for recovering this sparse signal and how bad is the solution going to be ?". The theorems that give answer to this question usually provide some bounds either pessimistic or loose and in some cases based on quantities, e.g., the Restricted Isometry Property, that cannot be easily computed in practice.

The formalism of the "state Evolution" is a fairly recent analysis framework that provides a lot of answers in a holistic manner and from a different perspective. As the name states it is based on the so-called "state" of the recovery algorithm, i.e., a parameter that describes accurately the course of the solution and whether the algorithm will succeed in solving the specific problem or not. The major difference between the state evolution and previous work is that it depends greatly on the sampling matrix (so far solid results regard random Gaussian matrices) and the that different results hold for different types of sparse signals, i.e., signals with their non-zero entries being only positive, or positive and negative and combination of thereof. The state evolution has been found to hold for the AMP algorithm and

several other of its renditions which will not occupy this text.

9.3.1 The Scalar Case

The analysis of the AMP algorithm depends a great deal on the scalar case analysis where

$$y = x + w, \tag{9.14}$$

is the measured signal x with additive noise w of zero mean and variance σ^2. This commences with the following,

$$\hat{x}(y) = \eta(y, \lambda) = \begin{cases} y - \lambda, \text{ if } y > \lambda \\ y + \lambda, \text{ if } y < -\lambda \\ 0, \text{ otherwise.} \end{cases}, \qquad \lambda = \theta \tag{9.15}$$

with the simple scalar estimator and the equivalence relationship for λ. This agrees with the previous discussion about this relationship and the avid reader is redirected to [69]. Furthermore the author in [69] points out that since the estimator $\hat{x}$ is basically a de-noising function, to scale the threshold with the noise level so the following is adopted

$$\theta = \alpha\sigma.$$

Another important definition is that of the *worst-case distribution* for an estimator, which is that distribution for the signal x to be estimated so that the Mean Square Error (MSE) is maximised. This has a very useful intuitive meaning; since the actual distribution of the signal is unknown then it makes sense to perform analysis based on the worst possible distribution. It turns out that for the scalar estimator in Equation (9.15) the worst case distribution is the following

$$p^{\#}(x) = \frac{\epsilon}{2}\delta_{-\infty} + (1 - \epsilon)\delta_0 + \frac{\epsilon}{2}\delta_{+\infty}$$

which is basically the sum of three Dirac distributions placed at $\pm\infty$ and 0 with probabilities $0.5 \cdot \epsilon$ and $1 - \epsilon$ respectively.

The worst-case MSE for the estimator can be computed in a straightforward manner,

$$\begin{gathered} \mathrm{E}\left[(\hat{x}(y) - x)^2\right] = \mathrm{E}\left[(\hat{x}(y) - x)^2 | x = -\infty\right] p(x = -\infty) \\ + \mathrm{E}\left[(\hat{x}(y) - x)^2 | x = 0\right] p(x = 0) + \mathrm{E}\left[(\hat{x}(y) - x)^2 | x = +\infty\right] p(x = +\infty) \end{gathered}$$

Breaking this down to the individual parts,

$$\begin{aligned}\mathrm{E}\left[(\hat{x}(y)-x)^2|x=-\infty\right]p(x=-\infty) &= \mathrm{E}\left[(\hat{x}(y)-x)^2|x=+\infty\right]p(x=+\infty)\\ &= \frac{\sigma^2}{2}\mathrm{E}\left[(w+\alpha\sigma)^2\right] = \frac{\epsilon}{2}\sigma^2(1+\alpha^2).\end{aligned}$$

For the term in the middle,

$$\begin{aligned}\mathrm{E}\left[(\hat{x}(y)-x)^2|x=0\right]p(x=0) = (1-\epsilon)[&\mathrm{E}\left[(w-\alpha\sigma)^2|w>\alpha\sigma\right]p(w>\alpha\sigma)+\\ &\mathrm{E}\left[(w+\alpha\sigma)^2|w<\alpha\sigma\right]p(w<\alpha\sigma)].\end{aligned}$$

Each of these two terms breaks down into three integrals,

$$\begin{aligned}\int_{-\infty}^{-\sigma\alpha} w^2\mathcal{N}(0,\sigma^2)dw &= \sigma^2(\alpha\phi(\alpha)+\Phi(-\alpha))\\ 2\alpha\sigma\int_{-\infty}^{-\sigma\alpha} w\mathcal{N}(0,\sigma^2)dw &= -2\alpha\sigma\phi(\alpha)\\ \alpha^2\sigma^2\int_{-\infty}^{-\sigma\alpha}\mathcal{N}(0,\sigma^2)dw &= \alpha^2\sigma^2\Phi(-\alpha).\end{aligned}$$

where $\phi(x)$ and $\Phi(x)$ are standard Gaussian density and cumulative functions respectively.

Combining all of the above, one then arrives at the expression for the worst-case MSE,

$$\sigma^2\cdot M(\epsilon,\alpha) = \sigma^2\left[\epsilon(1+\alpha^2)+2(1-\epsilon)(1+\alpha^2)\Phi(-\alpha)-\alpha\phi(\alpha)\right].$$

Notice that this intuitively scales with the noise variance just like θ.

Optimising the expression above with respect to α and substituting the resulting optimal $\alpha^{\#}(\epsilon)$ in the expression above, the *soft-threshold minimax risk* expression is formed,

$$M^{\#}(\epsilon) = M(\epsilon,\alpha^{\#}(\epsilon)). \tag{9.16}$$

The term *minimax* stems from the fact that the estimator that uses $\alpha^{\#}$ achieves the smallest possible MSE for the worst-case distribution. The expression for $\alpha^{\#}(\epsilon)$ does not have a closed form expression but it can be computed numerically quite easily as the solution of

$$M'(\epsilon,\alpha) = 2\epsilon\alpha+4(1-\epsilon)\left[\alpha\Phi(-\alpha)-\phi(\alpha)\right] = 0. \tag{9.17}$$

9.3.2 Heuristic Derivation of the State Evolution

The so-called *state evolution* framework is a way to accurately describe the evolution of the iterations of the AMP algorithm in the large system limit. The "state" of the algorithm refers to a single parameter with which the iterations can be fully described. The derivation of the state can be done heuristically by altering the AMP iterations as follows

$$\boldsymbol{x}_{t+1} = \eta\left(\boldsymbol{x}_t + \boldsymbol{H}_t^T \boldsymbol{z}_t, \theta_t\right) \tag{9.18}$$

$$\boldsymbol{z}_t = \boldsymbol{y}_t - \boldsymbol{H}_t \boldsymbol{x}_t. \tag{9.19}$$

In this fictitious setting, matrix $\boldsymbol{H}$ is re-sampled at each iteration. This implicitly means that a $\boldsymbol{y}_t = \boldsymbol{H}_t \boldsymbol{x}_t + \boldsymbol{w}$ is acquired afresh at each iteration. From the second equation the correction term was also removed for reasons that will become relevant shortly.

A single iteration of the equations above gives

$$\boldsymbol{x}_{t+1} = \eta\left(\boldsymbol{x} + \boldsymbol{H}_t \boldsymbol{w} + \boldsymbol{B}_t(\boldsymbol{x}_t - \boldsymbol{x}), \theta_t\right) \tag{9.20}$$

where $\boldsymbol{B}_t = \boldsymbol{I} - \boldsymbol{H}_t^T \boldsymbol{H}_t$. Focusing only on the main results [69], it is discovered that the following hold for the respective entries,

$$\left(\boldsymbol{H}_t^T \boldsymbol{w}\right)_i \sim \mathcal{N}(0, \sigma^2)$$

$$\left((\boldsymbol{B}_t(\boldsymbol{x}_t - \boldsymbol{x}))_{ij} \sim \mathcal{N}\left(0, \frac{\tilde{\tau}_t^2}{\delta}\right)\right.$$

where $\tilde{\tau}_t^2 = \lim_{m\to\infty} \frac{1}{m}\|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2$. Based on this it is then easy to see that the entries of the first argument of Equation (9.20) converge to

$$\boldsymbol{X}_0 + \tau_t \boldsymbol{Z} \qquad \text{with } \tau_t^2 = \sigma^2 + \frac{\tilde{\tau}_t^2}{\delta},$$

where $Z_i \sim \mathcal{N}(0, 1)$. Random variables $(X_0)_i$ are drawn from the distribution from which the original entries x_i are drawn. Based on the above derivations at iteration $t + 1$,

$$\tilde{\tau}_{t+1}^2 = \lim_{m\to\infty} \frac{1}{m}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}\|_2^2 = \mathrm{E}\left[\left(\eta(\boldsymbol{X}_0 + \tau_t \boldsymbol{Z}, \theta_t) - \boldsymbol{X}_0\right)^2\right]. \tag{9.21}$$

The temporal evolution of τ_t is known as the state of the algorithm and is given by the following equation,

$$\tau_{t+1}^2 = \sigma^2 + \frac{1}{\delta}\mathrm{E}\left[\left(\eta(\boldsymbol{X}_0 + \tau_t \boldsymbol{Z}, \theta_t) - \boldsymbol{X}_0\right)^2\right].$$

In a nutshell, τ_t was shown to summarise the two steps of the fictitious iterations in Equations (9.19). The relationships listed above are greatly based on the fact that matrix $\boldsymbol{H}$ is re-sampled at each iteration. As will be shown below, state evolution fails if this assumption is dropped. The additional correction term which was neglected in the beginning makes all the difference. This additional term causes the state evolution to be correct for the AMP algorithm.

In [7] the authors provide a powerful theorem for relating the MSE at each iteration of the AMP algorithm with the state of the algorithm as defined above. Without providing any further details the authors prove that under certain mild assumptions the following result holds,

$$\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} (x_i^{t+1} - x_i)^2 = \mathrm{E}\left[(\eta(\boldsymbol{X}_0 + \tau_t \boldsymbol{Z}, \theta_t) - \boldsymbol{X}_0)^2\right].$$

Taking a moment to appreciate this result, the theorem allows to predict the MSE of the algorithm at each iteration based on the state τ_t. Taking this intuition a bit further one can envision that this implies that at each iteration

$$\left(\boldsymbol{x}_t + \boldsymbol{H}^T \boldsymbol{z}_t\right)_i = x_i + \tilde{w}_i.$$

This is equivalent to saying that the estimates $\boldsymbol{x}_t$ are in reality individual estimates of the original signal entries with an increased noise level $\tilde{w}_i \sim \mathcal{N}(0, \tau_t^2)$. This comes by the name of *decoupling principle* since based on the developed theory the AMP acts as if m individual scalar problems are to be solved. The increased noise level is intuitively given by the state of the algorithm and relates to the "coupling" imposed by matrix $\boldsymbol{H}$.

Based on this intuition the threshold θ_t for the vector case ($\theta = \alpha\sigma$ in the scalar case) can be set equal to

$$\theta_t = \alpha\tau_t \approx \alpha \frac{1}{\sqrt{n}} \|\boldsymbol{z}_t\|_2$$

according to the equivalent noise level for the vector case. The temporal estimate of this can be given by the magnitude of the residual $\boldsymbol{z}_t$.

9.3.3 The Phase Transition Curve

The results listed above about the state evolution and its asymptotic convergence rely on the distribution $\boldsymbol{X}_0$ to make predictions. Going back to the discussion about the *soft threshold minimax risk* $M^{\#}(\epsilon)$, the authors in [31] provide yet another very useful result which relates this with the predicted one in Equation (9.21), $\tilde{\tau}_t^2$. Again,

presenting just the main results; if $\rho = \frac{\|\boldsymbol{x}\|_0}{n}$, $\epsilon = \frac{\|\boldsymbol{x}\|_0}{m} = \rho\delta$ and ρ_c is the solution of

$$\delta = M^{\#}(\epsilon)$$

then for every $\rho < \rho_c$ the following holds

$$\tilde{\tau}_{\infty}^2 = \frac{M^{\#}(\epsilon)}{1 - \frac{M^{\#}(\epsilon)}{\delta}}$$

while for every $\rho \geq \rho_c$

$$\tilde{\tau}_{\infty}^2 = \infty.$$

The corresponding values of ρ and δ at which this happens are given by

$$\rho(\alpha) = 1 - \frac{\alpha\Phi(-\alpha)}{\phi(\alpha)} \tag{9.22}$$

$$\delta(\alpha) = \frac{2\phi(\alpha)}{\alpha + 2(\phi(\alpha) - \alpha\Phi(-\alpha))} \tag{9.23}$$

in which case the optimal minimax threshold is used then $\alpha = \alpha^{\#}(\epsilon)$ as used in Equation (9.16). The expressions above result by solving Equation (9.17) for $\epsilon = \rho\delta$ and then factorising the resulting expression in two discrete factors.

To interpret this result the notion of the phase transition curve is introduced. Basically for every sparse recovery problem there is a problem size n, m and a sparse signal $\boldsymbol{x}$ with $\|\boldsymbol{x}\|_0 = k$ non-zero components. For every instance of such a sparse recovery problem, i.e., for every possible problem size and k, a recovery algorithm exhibits a certain degree of success. This means that if the algorithm is run over a fixed problem size and k for an infinite number of instances then based on some criterion (e.g., the MSE being smaller than a threshold) it will succeed with probability S. Using the notion of δ and ρ this means that for a given degree of success S the (δ, ρ) space is divided into two parts. The part for which the algorithm will succeed and the part for which it will fail; always based on the criterion which was chosen. This is the so-called phase transition, from the area of problems with (ρ, δ, k) for which the selected algorithm succeeds to the area of problems for which the algorithm fails.

What the result above provides is a closed form expression for the phase transition curve in parametric form - depending on the threshold α - for when the success is judged based on the MSE. For the AMP algorithm this means that running it for a range of values of δ and ρ, each time with $\alpha^{\#}(\epsilon)$ and by measuring success based on the MSE it is possible to draw the phase transition curve given the corresponding ϵ.

The following experimental setup is devised to demonstrate these results empir-

ically.

1. Divide the interval $[0.01, 0.99]$ in 50 equidistant points and form set $\mathcal{I}$.
2. Set the problem size $m = 1000$.
3. For each point $\delta \in \mathcal{I}$ do the following:
   - Set $m = \lceil \delta \cdot m \rceil$.
   - For each point $\rho \in \mathcal{I}$ do the following:
     (a) Set $s_\rho = 0$.
     (b) Set $k = \lceil \rho \cdot n \rceil$
     (c) Draw matrix $\boldsymbol{H} \in \mathbb{R}^{n \times m}$
     (d) Draw a sparse signal $\boldsymbol{x} \in \mathbb{R}^m$ with k non-zero entries placed at uniformly random indices and set uniformly random equal to ± 1.
     (e) Compute $\alpha^{\#}(\rho\delta)$ by solving Equation (9.17).
     (f) Run the AMP iterations in Equations (9.13) for 1000 iterations for $\alpha^{\#}(\rho\delta)$.
     (g) If $\frac{\|\boldsymbol{x}^{1000} - \boldsymbol{x}\|_2^2}{\|\boldsymbol{x}\|_2^2} < 0.1^4$ then set $s_\rho = s_\rho + 1$.
     (h) Repeat steps (c)-(g) 20 times.
     (i) Compute the empirical probability of success $S_{\delta,\rho} = \frac{s_\rho}{20}$.
4. Fit all the data points in $S_{\delta,\mathcal{I}}$ to a *generalised linear model* (see below) and compute the values $\rho_{50^{th}}$ at which the fitted probability of success is 50%.
5. For each point $\delta \in \mathcal{I}$ compute $\alpha^{\#}(\delta \cdot \rho_{50^{th}})$.
6. Compute the predicted values of $\rho(\alpha^{\#}), \delta(\alpha^{\#})$ from Equations (9.23).

In Step 4 of the procedure above the pairs $(\rho, S_{\delta,\mathcal{I}})$ are fitted to the following model using regression [29],

$$\text{logit}(S_{\delta,\mathcal{I}}) = a + b\rho.$$

where $\text{logit}(x) = \log \frac{x}{1-x}$ is the inverse of the logistic function. After the data pairs have been fitted to this model the point at which $S_{\delta,\mathcal{I}} = 0.5$ can be computed,

$$\rho_{50^{th}} = -\frac{\hat{a}}{\hat{b}}$$

with the values of the fitted regression parameters $\hat{a}, \hat{b}$.

The results of this long experiment are shown in Figure 9.4. The parametric curve computed with Equations (9.23) is plotted with the red thick line while the

empirical curve corresponding to the pairs $(\delta, \rho_{50^{th}})$ is drawn with the blue thin line. It is interesting to notice that the empirical curve showing the points at which the algorithm succeeds by 50% coincide with the predicted parametric points computed at the threshold $\alpha^{\#}$ used to achieve this performance.

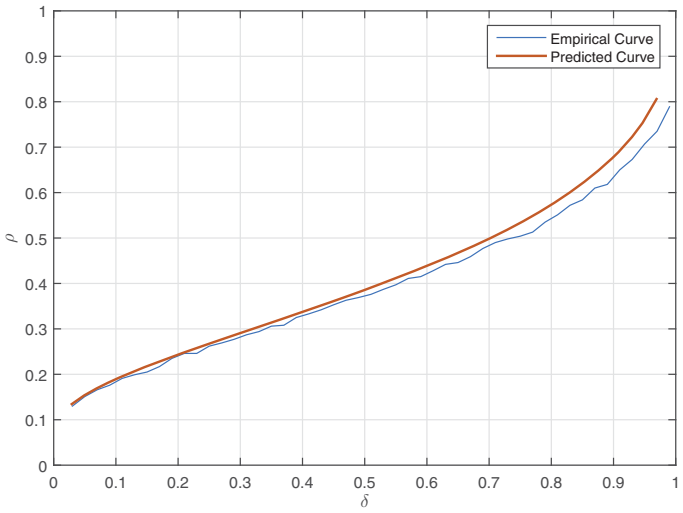


Figure 9.4: *Comparison of the empirical phase transition curve for the AMP algorithm and the predicted via state evolution.*

9.4 Relationship With Sparse Bayesian Learning

In Section 9.2 the route of deriving the AMP algorithm was shown to basically be an approximation as the name implies of an inference procedure based on distributions being interpreted as messages being exchanged between nodes on the corresponding factor graph. In Sparse Bayesian Learning which was presented in numerous occasions in this text inference is performed also in approximate manners by either the Type-II Maximum Likelihood procedure (Chapter 5) or by Variational methods (Chapter 7). Recall that the latter two methods were proven to be equivalent for *uninformative* distributions.

The signalling difference between the two methods of approximating inference is that the AMP does not require the computation of an inverse matrix while under the large system assumption, the quality of the estimates is improved via a specific prescription of the threshold to be used and the correction term in the computation of the residual $\boldsymbol{z}_t$ in Equations (9.13). There have been several attempts to marry the two approaches initially with [88] and later in [57]. There have also been empirical reports of cases with highly-coherent dictionaries [111] where SBL-based algorithms outperform several other approaches including the AMP by a great deal.

In the surrounding bibliography and community the theoretical results from the AMP analysis are indeed ground-breaking because of the state evolution frame-

work. It is yet uncertain what the relationship is between the two approaches. A central role in the analysis of the AMP algorithm plays the scalar estimator and its worst-case distribution. In this section an attempt is presented to understand the relationship between the *scalar SBL estimator* and the soft-threshold estimator as used in the AMP.

9.4.1 The Scalar SBL Estimator

From Equation (9.14) and the Sparse Bayesian Learning hierarchical model the following apply for the scalar case,

$$\begin{aligned} p(y|x) &= \mathcal{N}(x, \sigma^2) \\ p(x|\gamma) &= \mathcal{N}(0, \gamma^{-1}) \\ p(\gamma) &= \text{Gamma}(a, b). \end{aligned}$$

The Gamma hyper-prior in the last level of the model is set to what has been known so far in this text as the *uninformative* prior with $a = b = 0$. Following the steps of Chapter 5 the optimal value of γ is found as

$$\begin{aligned} \gamma^* &= \arg\max_{0<\gamma<+\infty} \log p(y|\gamma) \\ p(y|\gamma) &= \mathcal{N}(0, \sigma^2 + \gamma^{-1}), \end{aligned}$$

which results in

$$\gamma^* = \frac{1}{y^2 - \sigma^2}, \quad |y| > \sigma.$$

The scalar estimator is the mean of the posterior $p(x|y, \gamma) \propto p(y|x)p(x|\gamma)$ calculated at γ^*,

$$x(y) = \begin{cases} \frac{1}{\sigma^2\gamma^*+1} y = \frac{y^2-\sigma^2}{y^2} y, & |y| > \sigma \\ 0, \quad \text{otherwise.} \end{cases} \tag{9.24}$$

The graph in Figure 9.5 shows the comparison between the SBL estimator in Equation (9.24) derived above and the soft-threshold estimator as in Equation (9.15).

9.4.2 The SBL Worst Case Distribution

In Chapter 5 there has been a short discussion regarding the worst-case input signal the SBL inference algorithm can be given to recover. In the work of Wipf and his colleagues in [82], [108] and in more detail in [104] the worst-case scenario for the distribution of the sparse components is discussed from a different perspective. The authors pose this problem of finding the worst possible distribution for the sparse signal coefficients with respect to the corresponding cost function for SBL

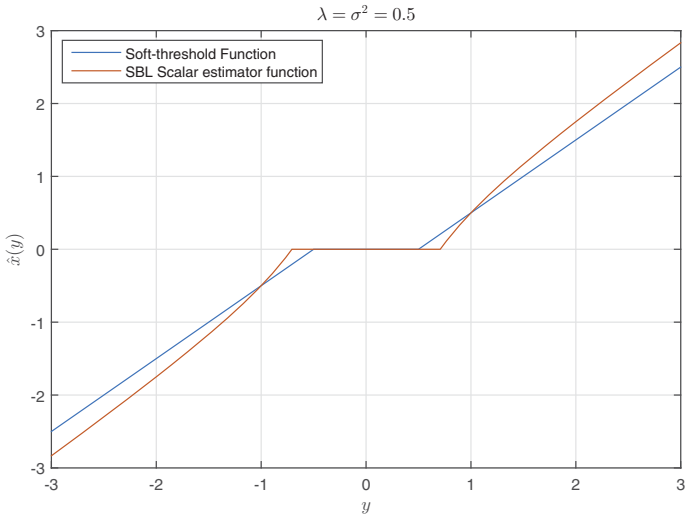


Figure 9.5: *Comparison of SBL and soft-threshold estimators for* $\lambda = \sigma = 0.5$.

(actually they consider a broader class which is more general but of the same form nonetheless), i.e., the one that will produce the most local minima. The result of that study is that the worst case for the cost function happens when all of the sparse signal coefficients are equal, e.g., $x_i = x_j = 1$. This result is backed up by a short discussion and examples but is not rigorously proven. The result is of high practical importance but not directly applicable in the discussion to take place in this text since care is taken in finding which is the worst possible distribution as far as the MSE is concerned. The relationship between the number and existence of local minima for the SBL cost function (recall that it is a non-convex function) and the MSE is not direct or straightforward to derive.

Moreover from what has been discussed so far for the State Evolution formalism, it is important to compare the worst case distribution of the SBL estimator with the MSE for the worst-case distribution for the soft-threshold estimator, given in Equation (9.3.1). For this purpose the three-point distribution is parametrised with the magnitude u and then that magnitude $\pm u^*$ which achieves the maximum distortion for the SBL scalar estimator is computed. The resulting three-point distribution will be the one maximising the MSE for the class of three-point distributions but it is not certain that this is the worst-case over all possible coefficient distributions. It serves as the means to compare the two estimators and what the performance of the scalar SBL estimator is when given a sparse signal drawn from a three-point distribution.

The MSE with respect to the considered 3-point distribution $p(x = \pm u) = \frac{\epsilon}{2}$ and $p(x = 0) = 1 - \epsilon$ is

$$M(u, \epsilon) = 2E[(\hat{x}(y) - x)^2 | x = u]p(x = u) + \mathrm{E}[(\hat{x}(y) - x)^2 | x = 0]p(x = 0).$$

After a tedious but straightforward calculation the resulting expression is,

$$\begin{aligned}
M(u,\epsilon) = 2\epsilon\Big[&\sigma^2\left((\sigma+u)\phi(\sigma+u,\sigma^2)+\Phi(-\sigma-u),0,\sigma^2)\right)+1\\
&-\sigma^2\left((\sigma-u)\phi(\sigma-u,\sigma^2)+\Phi(\sigma-u,0,\sigma^2)\right)\\
&+\int_{-\infty}^{-\sigma-u}\left(-\frac{2\sigma^2 w}{u+w}+\frac{\sigma^4}{(u+w)^2}\right)\mathcal{N}(w|0,\sigma^2)dw\\
&+\int_{\sigma-u}^{+\infty}\left(-\frac{2\sigma^2 w}{u+w}+\frac{\sigma^4}{(u+w)^2}\right)\mathcal{N}(w|0,\sigma^2)dw\\
&+u^2\left(-\Phi(-\sigma-u,0,\sigma^2)+\Phi(\sigma-u,0,\sigma^2)\right)\Big]\\
&+4\sigma^2(1-\epsilon)\left(\phi(1,1)-\Phi(-1,0,1)\right).
\end{aligned}$$

The following are defined as well,

$$\begin{aligned}
\phi(x,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}}\\
\Phi(x,\mu,\sigma^2) &= \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(w-\mu)^2}{2\sigma^2}}dw.
\end{aligned}$$

For future reference note that the expression for the MSE is linear with respect to ϵ as opposed to the expression for the soft-threshold estimator in Equation 9.16.

In order to derive the expression for u which maximises distortion the first order derivative $M'(u,\epsilon)$ has to be computed. After a long series of integrations this results in the following

$$\begin{aligned}
M'(u,\epsilon) = 2\epsilon\Big[&(\sigma+u)^2\phi(\sigma+u,\sigma^2)-(\sigma-u)^2\phi(\sigma-u,\sigma^2)+2\sigma^2\phi(\sigma-u,\sigma^2)\\
&-\left(2\sigma(\sigma+u)+\sigma^2\right)\phi(\sigma+u,\sigma^2)+\int_{-\infty}^{-\sigma-u}\left(\frac{2\sigma^2 w}{(u+w)^2}-\frac{2\sigma^4}{(u+w)^3}\right)\mathcal{N}(w|0,\sigma^2)dw\\
&-\left(2\sigma(\sigma-u)+\sigma^2\right)\phi(\sigma-u,\sigma^2)+\int_{\sigma-u}^{+\infty}\left(\frac{2\sigma^2 w}{(u+w)^2}-\frac{2\sigma^4}{(u+w)^3}\right)\mathcal{N}(w|0,\sigma^2)dw\\
&+2u\left(-\Phi(-\sigma-u,0,\sigma^2)+\Phi(\sigma-u,0,\sigma^2)\right)+u^2\left(\phi(-\sigma-u,\sigma^2)+-\phi(\sigma-u,\sigma)\right)\Big]
\end{aligned} \tag{9.25}$$

Solving the equation $M'(u,\epsilon)=0$ results in the worst-case $\pm u^*$ for the maximum MSE for the SBL scalar estimator. This will be referred to as

$$M^*(\epsilon) = M(u^*,\epsilon).$$

This quantity denotes the worst-case MSE the SBL scalar estimator produces when the sparse signal is drawn from the corresponding three-point distribution. The

value of

$$u^* \approx 2.16$$

is computed numerically since an analytic solution is intractable.

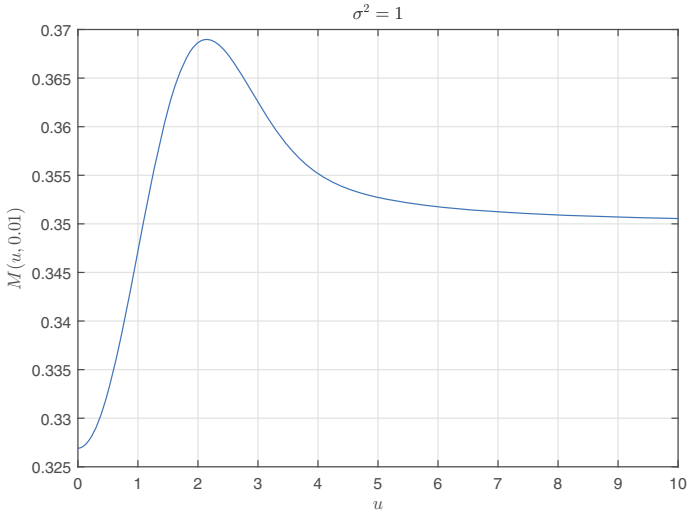


Figure 9.6: *MSE for the SBL scalar estimator with respect to u for $\epsilon = 0.01$ and $\sigma^2 = 1$.*

From Figure 9.6 it is evident that the SBL scalar estimator has a special behaviour as far as the worst-case three point distribution. Actually it exhibits a maximum at a point different than $+\infty$ at which it the estimator saturates producing a *smaller* MSE. By inspecting the expression for the computation of u^* in Equation (9.25) one can see that this point is of course dependent on the noise variance non-linearly. This is not the case with the soft-threshold estimator and the MSE given in Equation (9.16).

In order to get a sense of comparison between the two the MSE for the three-point distribution at $u = \pm\infty$ for the SBL scalar estimator is computed. Again, this is expected to be smaller than the case where $u = \pm u^*$. A similar series of computations can result in the MSE for when $u = \pm\infty$ so that a more close comparison is possible between the SBL estimator and the worst-case for the soft-threshold estimator. The expression for this is given below,

$$M^{\pm\infty}(\epsilon) = \epsilon + 4(1-\epsilon)\left(\phi(1) - \Phi(-1)\right).$$

The above quantity gives the MSE of the SBL estimator when the input signal is drawn from the worst-case distribution *for the soft-threshold estimator*.

Comparing the curves for the MSE's in Figure 9.7 one can easily understand that there is a specific range for the sparsity level ϵ for which the SBL estimator will produce smaller MSE when recovering signals drawn from the worst-case distribution for the soft-threshold estimator. Observing this from the opposite direction it is

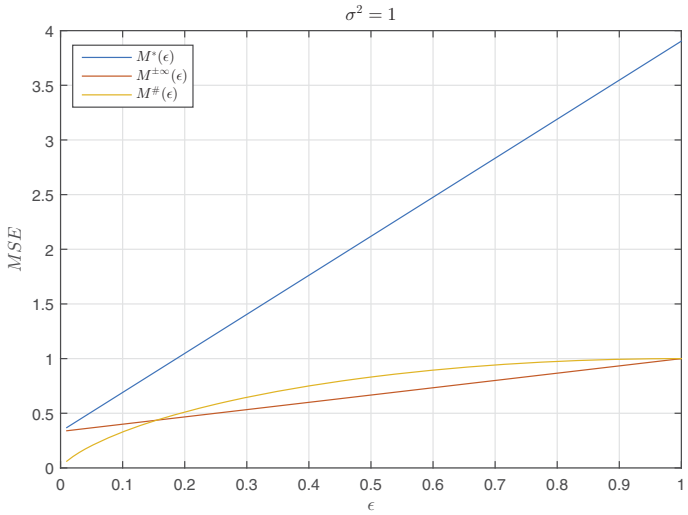


Figure 9.7: *MSE comparison for between the SBL estimator for* $u = \pm u^*, u = \pm\infty$ *and the soft-threshold estimator for its worst-case distribution. The assumed noise variance is* $\sigma^2 = 1$.

always certain that for the case where u is finite, for example $u = u^*$, the soft-threshold estimator will always perform better than the SBL estimator.

Another aspect worth considering is that the worst-case MSE for the SBL estimator is linear with respect to the sparsity level. This is easy to verify since u^* is independent of ϵ. There is a dependency of u^* on the noise level which means that the worst-case MSE will not scale linearly with respect to the noise level as opposed to the worst-case MSE for the soft-threshold estimator. Recall from Equation (9.17) that $\alpha^{\#}(\epsilon)$ depends linearly on both ϵ and σ^2 hence the worst-case MSE will not scale linearly with ϵ. From these two facts one can see a trade-off between knowing the sparsity level for the soft-threshold estimator and the SBL estimator which is agnostic.

9.5 Conclusion

In this chapter the very important framework of the AMP algorithm was presented along with the infamous State Evolution formalism. The purpose was not a deep analysis but merely an introduction towards establishing the relationship between the AMP and the SBL and more specifically what happens in the initial stages by assuming scalar estimators and their corresponding worst-case distributions.

Chapter 10

AMP-based Spectrum Analyser on an FPGA

In this chapter a fully working wideband spectrum analyser based on compressed sensing is presented. It is showcased how novel sampling techniques can be leveraged to greatly simplify the structure of such an apparatus and more specifically its analogue front-end without using *non-exotic* components. A basic assumption for such feats is that the radio frequency (RF) spectrum exhibits low occupancy at any given time instant. Basic motivation for this is the lack of hardware implementations despite the maturity of the theory and the analysis of the related algorithms.

The work carried out has revealed many practical aspects of these novelties that would otherwise have remained hidden behind the surrounding theory of compressed sensing. The limitations that have been uncovered can be considered as a starting point for future research. These include hardware considerations that reveal significant differences between theory and practice and hinder actual performance. Some crucial characteristics of the algorithms are pointed out.

A fully working prototype was produced, which implements a state-of-the-art random sampling scheme and an efficient reconstruction algorithm. More specifically the analogue signal is sampled using the Discrete Random Sampling theory [63] while recovery is performed by the Approximate Message Passing (AMP) algorithm [32]. Both are implemented on an FPGA in the form of a standalone core.

10.1 Technical Context

Conventional digital wideband spectrum sensing approaches usually monitor the spectrum using a swept, narrow bandwidth super-heterodyne receiver. These receivers dwell for a fixed period of time on a particular frequency band searching for transmissions before tuning to the next band in a pre-programmed list. Covering a

wide spectrum bandwidth is not instantaneous and it is possible that short duration transmissions will be missed. Other techniques that avoid such shortcomings are the ones that rely on filter-banks which can lead to increased hardware complexity of the analogue front-end.

The proposed spectrum analyser relies on unconventional sampling techniques that have been theoretically proven to achieve good performance based on sparse signal sampling and recovery methods without resorting to any of the solutions described above. The theory behind compressed sensing requires only a simple analogue front-end and a limited number of samples in exchange of a non-linear process for recovery instead of the classic interpolation formula. Within this project this was circumvented by employing computationally efficient algorithms that are amenable to FPGA hardware implementations.

The theory behind compressed sensing dictates that the analogue signal has to be sampled with a true random operator. Maybe one of the first examples is the Single-pixel camera [35] which employs a special laboratory setup to sample an image. The nature of imaging allows for this setup to exist and perform well up to a certain degree. A device which was presented in an earlier chapter is the Modulated Wideband Converter [68] and is aimed at compressively sampling wideband spectra. This technique was developed using bespoke hardware and DSP algorithms and is probably the first of its kind. It highly resembles sub-band processing devices with certain unique touches for adopting randomness and sparse recovery. Another implementation is that of [80] which is based on the Random Demodulator, also presented earlier. The researchers presented a prototype which worked for frequencies up to several hundreds of KHz.

For the prototype presented here the technical characteristics aim at achieving true RF wideband operation between $0.4 - 1.6$ GHz. Based on the initial requirements study it is speculated that 50% reductions in the sampling rate are possible with any further reductions only being limited by the existing commercial hardware. Another basic requirement is to have a very simple hardware analogue front-end as opposed to other compressed sensing prototypes. The wideband requirement places significant constraints on the implementation of such techniques with off-the-shelf components.

The two main steps in a technique as such are; first to acquire a reduced number of samples in a compressive manner and second to reconstruct the spectrum of the compressively sampled signal. In order to recover the original signal from its samples a non-linear process has to take place. This process, usually in the form of an iterative algorithm, recovers the original signal. For all practical purposes one cannot hope for exact recovery. The chosen algorithm can greatly affect the recovery accuracy and performance of the final system. The class of algorithms which

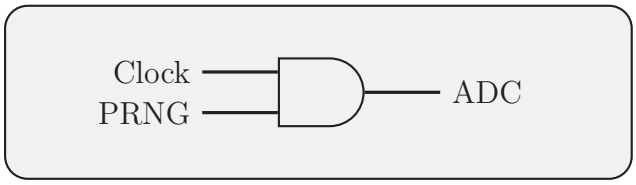


Figure 10.1: *Discrete Random Sampler gated clock.*

achieves the best recovery performance for the least number of samples is that of Linear Programming algorithms. Usually such algorithms require an abundance of computational power. A class of greedy algorithms such as the OMP, CoSaMP, SP provide a good trade-off between complexity and performance [91, 72, 26]. The downside of these algorithms is that they are not easily amenable to hardware implementations. The reason is that they require operations on sets of integers and computation of inverse matrices something which is more suitable for Digital Signal Processor implementations.

10.1.1 A Discrete Random Sampler

In order to implement a *true* compressive sampler randomness has to be introduced in the analogue domain. This would require exotic hardware or extremely bespoke solutions like the ones mentioned in Chapter 3. This path is not followed for the sake of simplicity of using off-the-shelf components. For this reason the Discrete Random Sampling theory is employed [63]. This technique dictates that samples are taken at randomly selected uniform clock pulses. This means a discretisation of a true non-uniform sampler which to the author's knowledge is a non-existent device or at least not yet commercially available.

The random sampler quite simply consists of an Analogue-to-Digital converter clocked by a random clock. The random clock is actually an AND-gated clock pulse with the second input to the gate being the output of a pseudo-random number generator (PRNG). This is shown in Figure 10.1.

A very important aspect of this sampler is that the minimum sample-spacing [40] for the ADC corresponds to the maximum frequency to be captured by the sampler. This is clarified in the Figure 10.2 diagram showing the relationship between the two clocks. For the proposed wideband spectrum analyser a bandwidth of $0.4 - 1.6$ GHz is aimed for hence the uniform clock behind the random sampler has to be able to support this bandwidth.

10.1.2 AMP for Sparse Recovery

Recently, theory has shown that a class of computationally inexpensive algorithms is possible to reach the performance of ℓ_1 optimisation under certain large

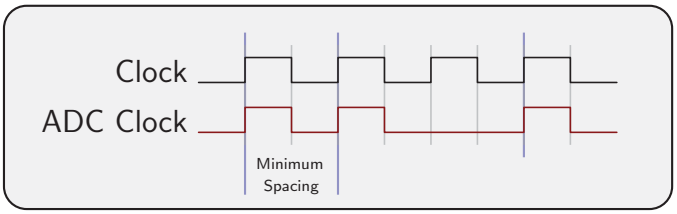


Figure 10.2: *ADC Random Clock. Notice minimum sample spacing. Blue lines represent instants at which the PRNG select a uniform clock pulse for capturing a sample with the ADC.*

system limits [32]. The AMP algorithm has been proven to exhibit some very favourable attributes, i.e., it does not require any matrix inverse operations or set operations ans is only based on simple matrix-vector operations. Moreover this class of algorithms has been theoretically proven to achieve performance close to that of Linear Programming in the large system regime. Some hardware implementations have been reported but differ significantly from the technique proposed here for wideband RF spectrum recovery [5].

To summarise, the two main technical challenges are:

1. The implementation of an efficient random sampler to compressively sample the signal. The sampler needs to be simple and relatively unbiased.

2. The implementation of a hardware-friendly algorithm to reconstruct the compressively sampled signal. The chosen algorithm will be empirically tested.

10.2 System Architecture

In this section the overall system architecture is presented. At first the *ideal system architecture* is discussed and then the reasons why one must resort to the *actual system architecture* used in the implementation.

10.2.1 Ideal System Architecture

In Figure 10.3 the functional diagram of the system is shown. The yellow and blue boxes along with the AND gate and the ADC are what consist the random sampler. The randomly generated digital samples are fed into the reconstruction algorithm core which implements the AMP algorithm. The term "Low-rate ADC" refers to the fact that the effective number of samples used is far lower than the Nyquist rate in the average.

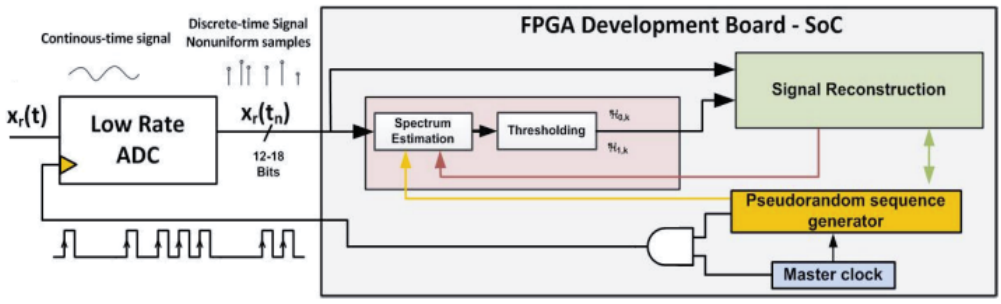


Figure 10.3: *Ideal System Architecture.*

10.2.2 Actual System Architecture

The chosen ADC device is capable of supporting a minimum discrete random sample spacing for frequencies up to 1.6 GHz and outputs samples for two channels (labelled In-phase and Quadrature) in a time-interleaved manner so as to reach the advertised bandwidth. Figure 10.4 shows the state at the data bus of the ADC for two consecutive clock cycles at t and t-8. The ADC data buses are 4 in total and are 8-bit wide. These are the I and Q lines and their delayed versions I_d and Q_d for the two channels respectively.

Clock Edge	Q_d	I_d	Q	I
$\uparrow$	$Sample[t]$	$Sample[t-1]$	$Sample[t-2]$	$Sample[t-3]$
$\downarrow$	$Sample[t-4]$	$Sample[t-5]$	$Sample[t-6]$	$Sample[t-7]$
$\uparrow$	$Sample[t-8]$	$Sample[t-9]$	$Sample[t-10]$	$Sample[t-11]$
$\downarrow$	$Sample[t-12]$	$Sample[t-13]$	$Sample[t-14]$	$Sample[t-15]$

Figure 10.4: *ADC data output timing.*

The ADC architecture impairs the way theory is applied to support the ideal system architecture in Figure 10.3. In short, one cannot randomly trigger the ADC to acquire one single sample but a group of 8 samples instead. Furthermore the ADC does not provide a stable mode of operation on a random clock but its operation is driven by an inhibit function communicated via a serial command. This complicates even further the direct adaptation of theory into practice. To circumvent this issue it was decided that a buffer to be implemented for the ADC samples and then the subsequent random clocking to be performed offline once a sufficiently large number of samples is acquired. For instance if the subsequent processing is to be performed on a Fourier grid of 4096 samples then a buffer of 4096 samples is implemented since the output of the PRNG is taken to be unbiased (equal number of zeros and ones).

This behaviour simulates the theoretical device described in the previous subsection.

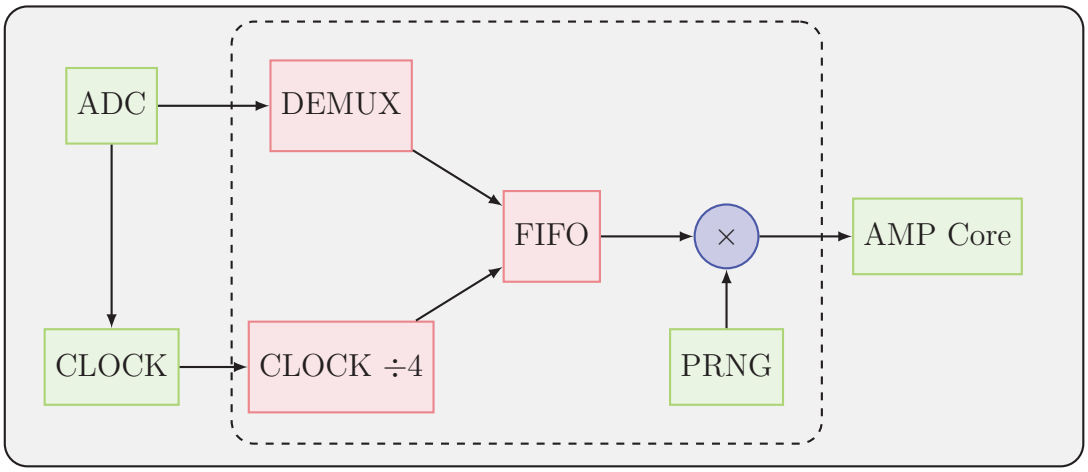


Figure 10.5: *Actual System Architecture.*

The diagram in Figure 10.5 is a functional description of the actual system. The blocks in red are the ones needed for the system to function given the specific ADC device. The red de-multiplexer block does not introduce any latency in the design and is a requirement due to the ADC output sample order.

The following actual numerical values describe the data rates in the system:

- ADC Clock : 1520 MHz
- ADC Data rate : $^{2 \times 1520}/_{8} = 380$ Samples/Second
- PRNG bit rate : 100 Mhz (FPGA device clock)
- AMP Core : 100 Mhz core clock

The data rate into the AMP Core is dependent on the processing rate of the core. Blocks of 4096 samples are mixed with the PRNG and presented to the AMP Core when it has finished processing a previous block. This rate will be substantially lower than the ADC sample rate.

The limitations underscored above regarding the ADC architecture are key into understanding why this technique will be difficult to implement even with future and more advanced ADC devices.

1. Any ADC able to achieve the minimum sample spacing for random sampling will also be able to achieve the entire bandwidth. Then we face the question whether any sparse processing algorithm will be able to achieve better results than traditional DSP algorithm both in terms of speed and accuracy.

2. In order to meet the high demands in data rates between the ADC and any other device FPGA/ASIC, the data transfers will definitely not be one sample

at a time. This fact along with the ADC's time-interleaved (possibly) construction implies some sort of pipeline which means that random selection of one single sample will not be possible. Currently there is no supporting theory for randomly sampling a signal in sets of samples instead of one sample at a time.

3. Any gains from the AMP core will have to come from sparse processing of the signal either in terms of accuracy or via a Sparse FFT algorithm (currently existing in an experimental setting) being able to outperform a traditional FFT core in terms of complexity.

10.3 The Approximate Message Passing Core

The AMP algorithm belongs in the class of sparse signal recovery algorithms that employ a soft threshold function to select the active components in the signal. In our case the Fourier basis will be considered hence the dominant frequencies in the sampled signal are iteratively being recovered until convergence to a certain margin or a predefined number of iterations has been reached. The AMP can be easily implemented in hardware since there are no explicit computations of the inverse of a matrix involved with the workhorse being the matrix-vector multiplication operation.

The AMP core for the FPGA utilises an FFT core to implement the matrix-vector multiplication operation since the considered compressed sensing matrix is the partial Fourier matrix. This way, the AMP core complexity is governed by the complexity of the FFT core used in the implementation. This in general is highly optimised and provided by the FPGA vendor or other.

10.3.1 The AMP algorithm

It is assumed that the random sampler has produced a vector of samples $\boldsymbol{y}$ with dimension m. For the implementation it is assumed that the signal is going to be sparse in the Fourier basis of dimension N which is chosen according to the device capabilities. The basic steps of the AMP algorithm are shown in Algorithm 9.

Subscript i denotes the iteration number. Function $\eta(x, \theta)$ outputs the value of x if its absolute value is greater than the threshold θ and 0 otherwise. The dimension of $\boldsymbol{y}$ is taken to be half of the size of the FFT ($m = {}^{N}/_{2}$) since it is assumed that the PRNG is unbiased and that the subsampling rate is 50%. Matrix F is the Discrete Fourier matrix. The ℓ_0 norm at Step 3 simply counts the non-zero elements of the operand.

Algorithm 9 AMP Algorithm for FFT basis.

Initialise: $\boldsymbol{r}_0 = \boldsymbol{y}$, $\boldsymbol{s}_0 = \boldsymbol{0}$.

Iteration $i = 1..i_{max}$:

1. $\theta = \lambda \cdot \|\boldsymbol{r}_{i-1}\|_2 \cdot 1/\sqrt{m}$
2. $\boldsymbol{s}_i = \eta(\boldsymbol{s}_{i-1} + \boldsymbol{F}\boldsymbol{r}_{i-1}, \theta)$
3. $\boldsymbol{r}_i = \boldsymbol{y} - \boldsymbol{F}^T\boldsymbol{s}_i + \boldsymbol{r}_{i-1}\|\boldsymbol{s}_i\|_0 \cdot 1/m$

In Step 3 of the algorithm the forward Fourier transform of the residual signal $\boldsymbol{s}_i$ is taken and the result is being sub-sampled by pre-loading the PRNG with the same seed that was used when sampling the signal with the random sampler. In short, the FFT output samples are chosen based on the same pseudo-random sequence with which the analogue signal was sampled. This is a very efficient way of implementing the partial Fourier matrix transform.

Similarly in Step 2, the inverse Fourier transform of signal $\boldsymbol{r}_{i-1}$ is taken. In this step the signal $\boldsymbol{r}_{i-1}$ is zero-padded according to the PRNG after being pre-loaded with the same seed as before. Recall that vector $\boldsymbol{r}_{i-1}$ is of dimension m.

10.3.2 Functional Description of the AMP core

The steps in Algorithm 10 describe the actual operations of the AMP core. The actual finite state machines responsible for these are not described here.

In many algorithms for Compressed Sensing the stopping criteria varies. In most of the greedy pursuits the stopping criterion is that the mean squared error falls below a certain threshold. In iterative threshold algorithms like the AMP such a criterion can also be used. For all these algorithms convergence is tightly tied with the maximum iteration count. For this prototype it is was decided to use the iteration count as a stopping criterion as a safety measure in case the algorithm diverged, i.e., avoid an infinite hardware loop. As a matter of fact even if the MSE was used a maximum iteration count would also be used for this reason.

A decisive factor on the size of the FFT core to be used is the sub-sampling rate. A buffer has to be utilised and this means that for lower sub-sampling rates a longer window has to be used (i.e., longer buffer) to allow for proper implementation of discrete random sampling. This in turn means that lower average sampling rates dictate for a smaller FFT core (in order for the longer buffer and the AMP core to fit in the same FPGA). Based on the capabilities of the device a sub-sampling rate has to be chosen so that both the buffer and the AMP core to both fit in the FPGA.

Algorithm 10 Functional Description of the AMP Core.

Initialise: $\boldsymbol{r}_0 = \boldsymbol{y}$, $\boldsymbol{s}_0 = \boldsymbol{0}$.

1. Load PRNG with seed. Read the samples from the buffer into the AMP core based on the output of the PRNG.
2. Calculate $\|\boldsymbol{r}_0\|_2$ at the same time.

Iteration $i = 1..i_{max}$:

1. $\theta = \lambda \cdot \|\boldsymbol{r}_{i-1}\|_2 \cdot {}^1/\sqrt{m}$
   - Check for iteration count. Calculate θ before entering Step 2.
2. $\boldsymbol{s}_i = \eta(\boldsymbol{s}_{i-1} + \boldsymbol{F}\boldsymbol{r}_{i-1}, \theta)$
   - Load PRNG with seed.
   - Load $\boldsymbol{r}_{i-1}$ into FFT core based on the PRNG (zero-padding).
   - Unload results from FFT core and read $\boldsymbol{s}_{i-1}$ while computing the threshold function. Compute $\|\boldsymbol{s}_i\|_0$ at the same time for Step 3.
3. $\boldsymbol{r}_i = \boldsymbol{y} - \boldsymbol{F}^T\boldsymbol{s}_i + \boldsymbol{r}_{i-1}\|\boldsymbol{s}_i\|_0 \cdot {}^1/m$
   - Load $\boldsymbol{s}_i$ into FFT core.
   - Load PRNG with seed.
   - Unload FFT results from core based on the PRNG output (partial FFT).
   - Calculate $\|\boldsymbol{r}_i\|_2$ at the same time required in Step 1 for iteration $i + 1$.

10.3.3 Limited Numerical Precision

All arithmetic operations are carried out in 2's complement and the chosen accuracy of the fixed point arithmetic used is 24 bits. This is a design parameter and can be defined before synthesis of the core. In all cases the numerical accuracy has to accommodate the scaling introduced by the FFT core and the bit length of each mathematical operation.

By observing the steps of the algorithm it is evident that the main source of significant bit growth comes from the FFT core and the inherent butterfly structures. The worst-case scenario for this bit growth for a Radix-2 FFT is,

$$\text{input length} + \log_2(\text{transform length}) + 1.$$

Assuming a 24-bit word length and a transform length of 4096 the resulting FFT samples would require 37-bit long words. A very conservative scaling schedule can be employed for each butterfly stage but approach is not followed here.

The numerical performance of an algorithm in fixed point arithmetic is a research subject on its own and some of its aspects will be verified in the section to follow. In order for the AMP algorithm to remain friendly towards any hardware platform

it is essential to be able to perform well in fixed point arithmetic.

10.4 Experimental Results

To assess the performance and validity of the results the extensive tools provided by the Xilinx ISE and the Modelsim simulator are used. The bit–accurate model of the Xiling FFT core is also used to build a model of the AMP core on MATLAB.

In the tests with the simulator a test-bench was written with all the necessary signals and output. The output of the random generator is firstly output to a file on the disk to get the random sequence of bits for 4096 clock cycles (assuming a 4096-sample FFT core). This sequence is then used in MATLAB to generate and randomly sample an exactly sparse signal with 4 active frequencies (placed at $100, 300, 600$ and 800 MHz) in the Fourier domain with a 4096 point DFT matrix. The precision of this signal is then quantised to 8 bits (ADC output word length) and written in the input file for the Verilog test-bench. The test-bench is then run for 10 iterations with two different values for λ. The test-bench is also run for two different cases; firstly where the amplitude of the active frequencies is chosen to be the same and secondly where the amplitude of the active frequencies is different and decreasing.

Using the bit-accurate model an empirical study of the algorithm's parameters is attempted. Considered also are test cases with spectral leakage in the frequency domain which corresponds to all practical cases of interest. Lower sub-sampling rates are also attempted, something impossible to implement on the device due to the hardware requirements.

10.4.1 Simulator Experiments

From the simulator it was possible to measure the exact time for a single AMP iteration at a device clock of 100 MHz and a transform size of 4096 samples to be,

$$1 \times \text{AMP iteration } = 4.6 \text{ ms}.$$

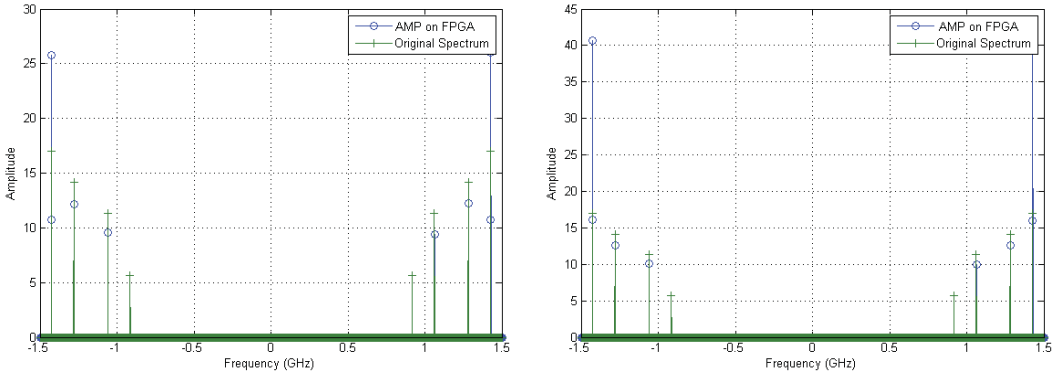
It should be noted that 4096 samples constitutes approximately 1.3 μs of data. Assuming an iteration count of 10, the algorithm is running at a 0.003% duty cycle. This will severely impact the ability to detect very short duration transmissions. However, for a one second burst of communications, the system will still sample this over 20 times resulting in a high probability of detection. This can be further improved by using more processing resources.

Currently, the duty cycle of this implementation is not competitive compared with state-of-the-art super-heterodyne receivers. However, the actual system archi-

tecture has more potential to improve with advances in processor technology than super-heterodyne receivers.

Experiment 1

In this test, the value of λ in the AMP algorithm is set to 15 and the AMP algorithm is run for 10 iterations. Below are the reconstructed spectra for iterations $4, 5$ and 10 compared against the original spectrum.



(a) **(b)**

Figure 10.6: *(a)* 4^{th} *iteration. (b)* 5^{th} *iteration. Regulariser* $\lambda = 15$.

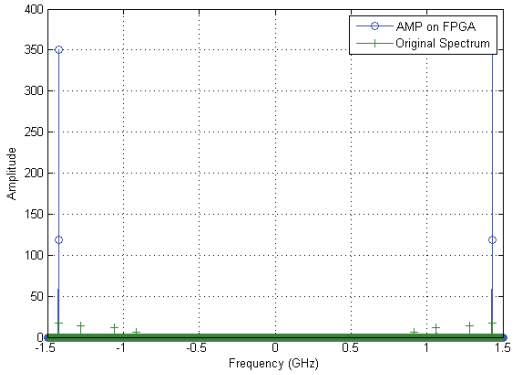


Figure 10.7: 10^{th} *iteration. Regulariser* $\lambda = 15$.

Experiment 2

The same experiment is run with the same value for λ but this time the simulated sparse signal is generated with equal amplitudes for the active frequencies.

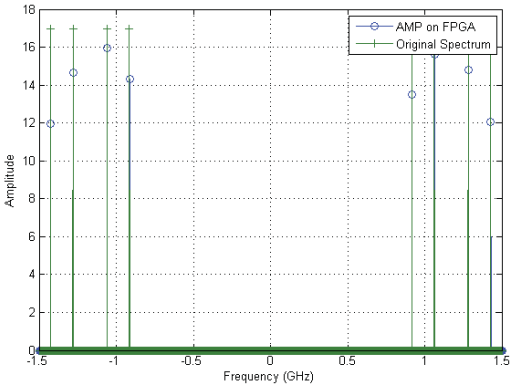
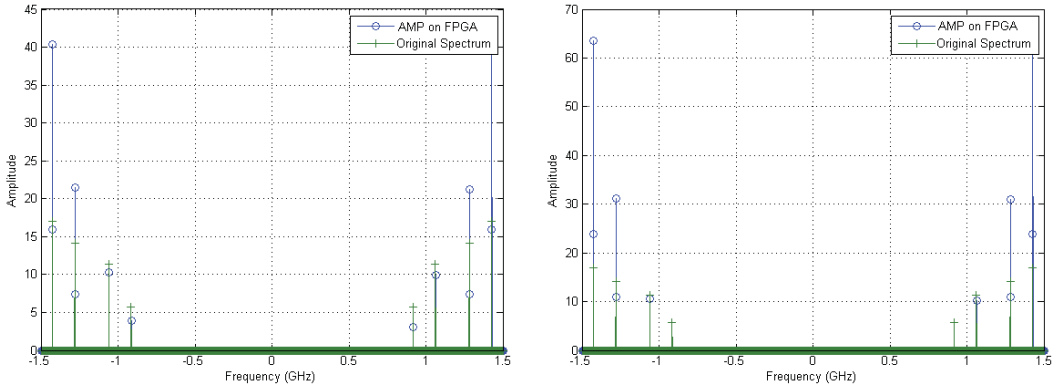


Figure 10.8: 5^{th} *iteration. Regulariser* $\lambda = 15$. *Equal Amplitudes.*

Experiment 3

The same experiment as Experiment 1 but the value of the regulariser is reduced to $\lambda = 5$.



(a) **(b)**

Figure 10.9: *(a)* 4^{th} *iteration. (b)* 5^{th} *iteration. Regulariser* $\lambda = 5$.

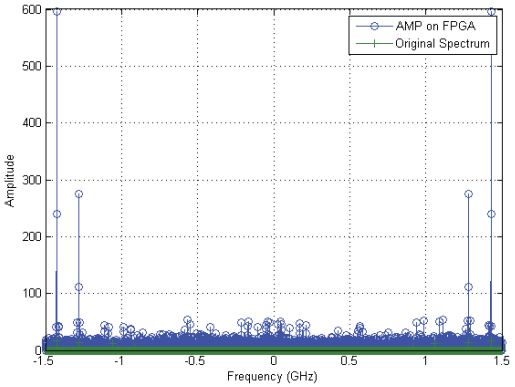


Figure 10.10: 10^{th} *iteration. Regulariser* $\lambda = 5$.

10.4.2 Number of Iterations

In theory the algorithm will converge to a stationary point at which the sparse signal will have been recovered within certain accuracy. The algorithm exhibits a steady-state at that point and any further iterations will have little effect on the result. In this particular implementation of the AMP algorithm as an FPGA core with fixed point arithmetic of limited precision, convergence to that stationary point is not always guaranteed before the algorithm's intermediate results go over the predefined word length.

In Experiment 1 it is observed that from iteration 4 to 5 the spectrum quality has improved with the frequency components being closer to the truth. In iteration 4 most of the spurious frequencies have been discarded by the algorithm as well. By progressing even further in iteration 10 it can be seen that the algorithm has diverged by recovering only one major component. In effect this means that some of the estimates have grown beyond the accuracy, hence they have been regularised out by the threshold function in Step 2.

By decreasing the value of λ in Experiment 3 it is possible to attain more of the components but the algorithm also diverges at high iteration counts simply by recovering non-dominant frequency components. In any practical setting the number of iterations has to be carefully chosen for the recovery algorithm to remain within the numerical accuracy.

10.4.3 Regulariser

The regulariser value defines the maximum frequency to be allowed during recovery. By comparing Experiments 1 and 3 it is recognised that all of the components have been identified by the algorithm in Figure 10.9 for $\lambda = 5$, along with some spurious components. In Experiment 3 for $\lambda = 15$ it was impossible to recover the component at 100 MHz with the smallest amplitude.

The value λ of the regulariser tells only half of the story for the threshold. In Step 1 of the algorithm the threshold is computed and it is directly proportional to the value of λ. The magnitude of the residual signal also plays an important role which is demonstrated by observing Experiment 2. In this case; all of the frequency components are chosen to have the same amplitude and the value of λ is set to 15 something which results in the component at 100 MHz not to be recovered. We see that when the frequency components have the same amplitude they are all recovered successfully by AMP at the 5^{th} iteration (Figure 10.8). This signifies that the relevant frequency amplitude plays an important role in recovery especially in fixed point applications. This means that a high amplitude component will result in a threshold value small enough to recover a component only up to a

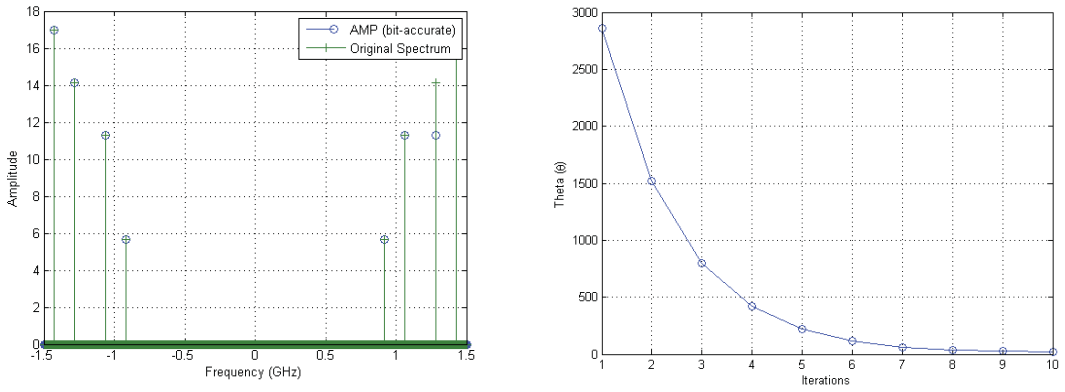
comparable amplitude. This phenomenon disappears in floating-point and double precision computations since the maximum iteration count can be larger without the algorithm diverging.

10.5 Bit-Accurate Model Simulations

In order to further explore the complications behind the issues regarding the regulariser in a controlled manner the FFT core bit-accurate model is used in MATLAB. The bit-accurate model of the core is provided in C++ which is then compiled into a MEX function. The model allows the numerical study of the AMP algorithm. The arithmetic functions described in the steps of the AMP algorithm are also simulated exactly by quantizing the results appropriately. By using the bit-accurate model it is also possible to simulate lower sub-sampling rates. As it was discussed earlier in the text, the sub-sampling rates on the actual device are limited by the size of the FFT used in the AMP and of course by the capabilities of the device. The performance of the AMP core at lower average sampling rates and/or with higher point FFT's can be assessed.

10.5.1 Regulariser

The same experiment as above is repeated, where the frequency components are of decreasing amplitude and placed exactly on the Fourier grid (4096 points). The bit-accurate algorithm is run for 10 iterations. During the course of the 10 iterations the values of the regulariser θ are stored. The value of $\lambda = 5$, is kept the same for this experiment.



(a) **(b)**

Figure 10.11: *(a) AMP with FFT bit-accurate model* 10^{th} *Iteration. (b) The value of* θ *during the* 10 *iterations.*

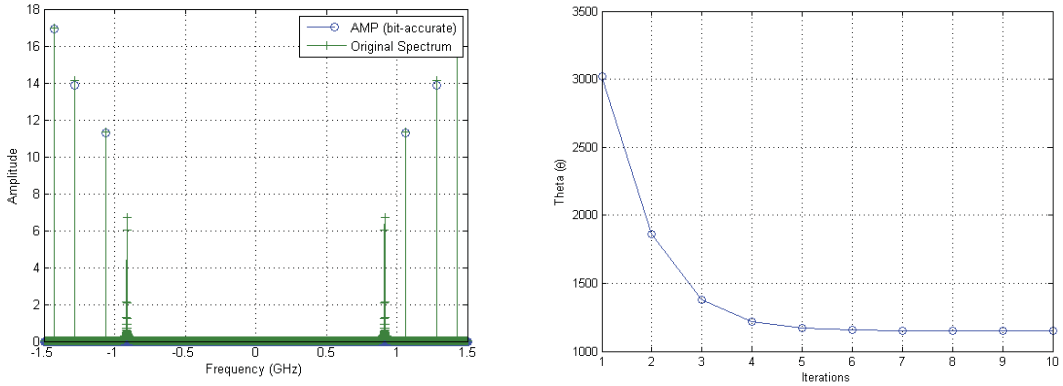
On the left-hand side of Figure 10.11 it is observed that the bit-accurate model of

the AMP produced to a very high degree the same spectrum as that of the original signal. Please note that the AMP has recovered many small-amplitude components not clearly visible on the plot.

The results presented above show that the AMP algorithm does not suffer from numerical instability issues as shown in Experiment 3. The bit-accurate model produced a nearly-perfect spectrum after 10 iterations without diverging something which was present in Experiment 3 in the Modelsim simulator after the same number of iterations and the same value for λ. This suggests either that the design suffers from timing issues (divisors, multipliers, adders) or that the AMP bit-accurate model lacks details from the actual HDL implementation.

10.5.2 Spectral Leakage

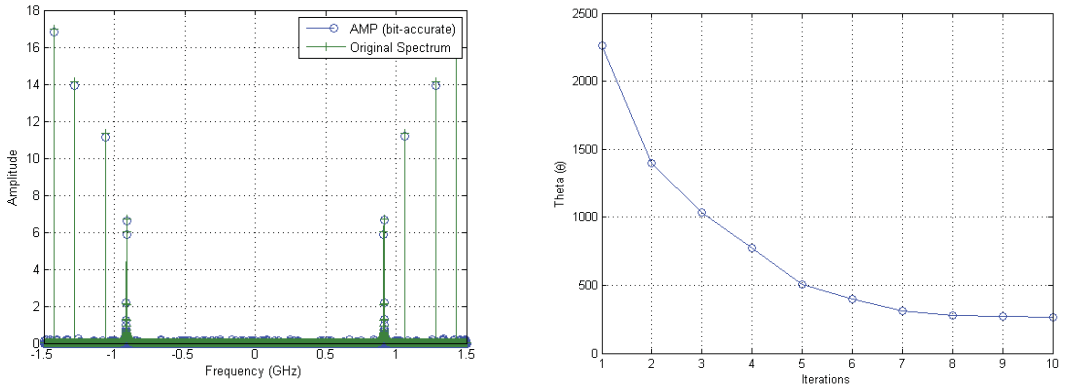
So far the experiments where performed on an exactly sparse spectrum which means that the amplitude components are placed exactly on the Fourier grid. In this experiment the frequency component at 800 MHz is replaced with one that does not exactly fit the Fourier spacing so that leakage is introduced. The value of λ is left the same as before.



(a) **(b)**

Figure 10.12: *(a) AMP with FFT bit-accurate model* 10^{th} *Iteration. (b) The value of* θ *during the* 10 *iterations.*

In Figure 10.12 we notice the additional spectral components in the original spectrum. At the 10^{th} iteration the algorithm was not able to recover any of the additional components. By studying the value of θ during the course of the algorithm it can be seen that the algorithm has converged to a larger value compared to the earlier example without any leakage. This means that the algorithm will have a larger threshold for the recovered frequencies resulting in many of them – including the desired ones – to be pruned out. The same experiment with a slightly decreased value of λ is repeated. The results are shown below.



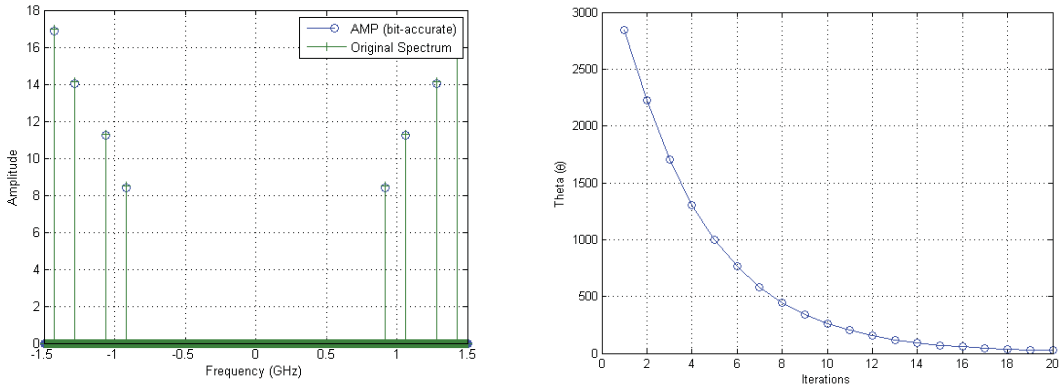
(a) **(b)**

Figure 10.13: *The same as in Figure 10.12 with a slightly smaller value for* λ.

The algorithm was able to recover many of the additional frequencies with a smaller value for the λ in Figure 10.13. By looking at the plot on the right-hand side, the regulariser value has reached a smaller threshold, hence allowing the desired frequencies to appear upon convergence. The last two experiments point out the importance of the regulariser during the execution of the algorithm. Should any numerical issues exist like finite word-length or overflows these will greatly affect the algorithms performance. After many tests there were no issues to report with the bit-accurate algorithm diverging or FFT overflows. The actual results from the algorithm's steps have been thoroughly checked with the Modelsim simulator. In the bit-accurate model, the actual behaviour of the dividers, multipliers and adders that are actually implemented in the AMP core was not modelled. This is another factor which might have affected the results in the actual experiments.

10.5.3 Sub-sampling Rate

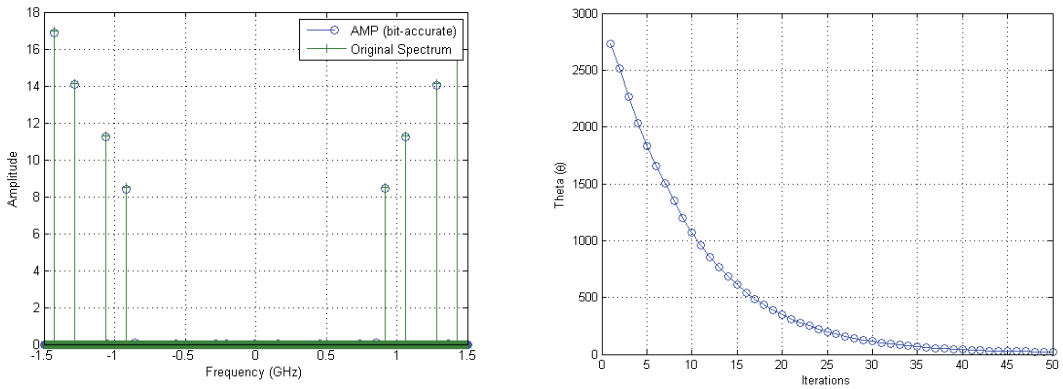
In this series of tests a short empirical study of the sub-sampling rate is attempted. In the experiments that have been discussed so far the rate was 50% meaning that a sufficiently long sample window was taken and the PRNG has chosen roughly half of the samples. The same experimental procedure is followed but lower sub-sampling rates are implemented by appropriately under-sampling the PRNG output which was recorded by running the Modelsim simulator. Below are the results for 25% sub-sampling rates.



(a) **(b)**

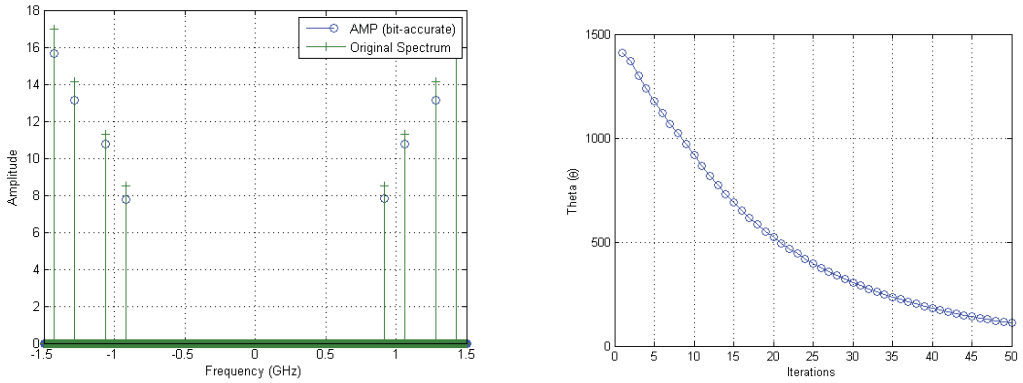
Figure 10.14: *Results for* 25% *sub-sampling rate.*

Comparing these results with the previous ones, the algorithm takes a larger number of iterations to converge but still manages to recover accurately the frequency components. Repeating the same experiment for lower average sampling rates:



(a) **(b)**

Figure 10.15: *Results for* 12.5% *sub-sampling rate.*

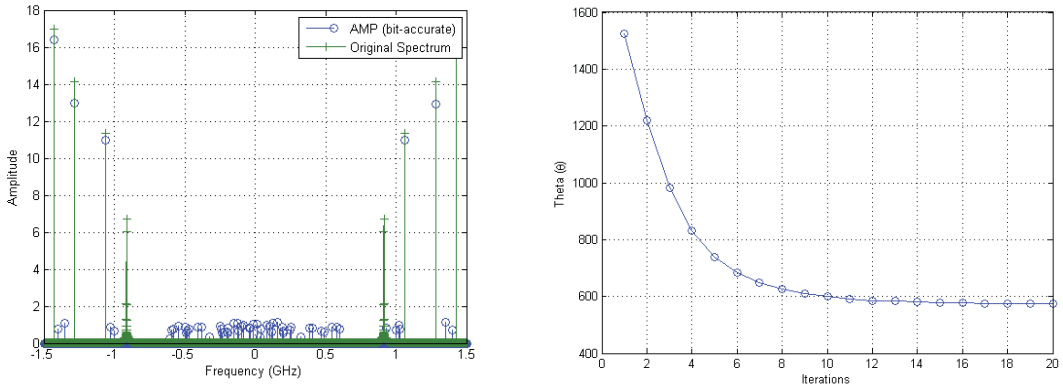


(a) **(b)**

Figure 10.16: *Results for* 6.25% *sub-sampling rate.*

The bit-accurate model produces exact spectra for up to 6.25% sub- sampling rate. The algorithm reaches convergence at a higher number of iterations.

In the following experiment leakage is introduced for 25% sub-sampling rate.



(a) **(b)**

Figure 10.17: *Results for* 25% *sub-sampling rate with spectral leakage.*

The algorithm converges without recovering any frequencies near the missing one. Some positive results have been reported during the tests by altering the value of λ but the spectrum was not sparse and resembled the one of Figure 10.17. Leakage plays an important role in recovering a sparse signal and it is not to be overlooked.

10.6 Field Trials

The prototype was exposed in a series of experiments to assess coverage and sensitivity with only the basic components of an analogue front-end, namely an omni-directional antenna and an amplifier. The choice of these two components was generic and no special study took place prior to their choice. The basic limitations of the system such as dynamic range were verified.

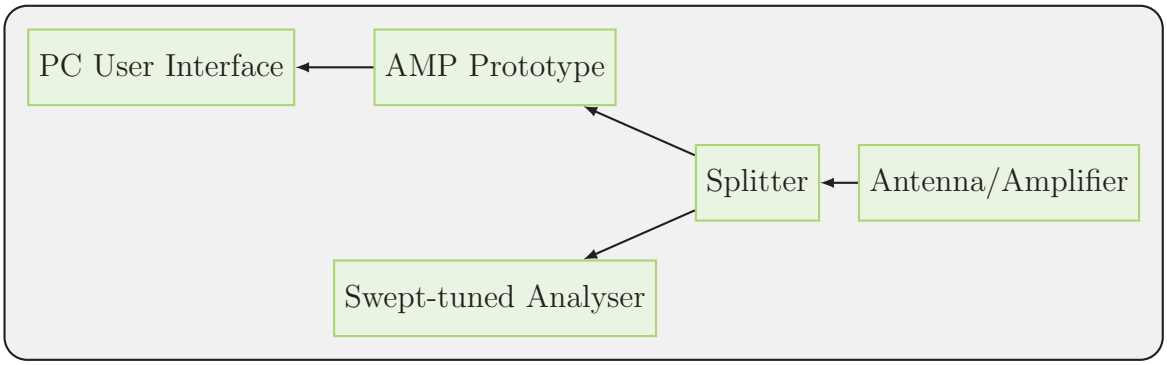


Figure 10.18: *Experimental setup signal path.*

Alongside with the evaluation module a swept-tuned Spectrum analyser was used and was fed with the input from the antenna with a splitter. For the purposes of the demonstration a graphical user interface was written in MATLAB. The interface was able to generate alarms when the power level on a specific band of the spectrum was present.

Several other limitations have been examined such as the algorithm's performance under limited numerical accuracy and fixed-point arithmetic. These also regard specific parameters of the algorithm which affect performance such as the number of iterations. These findings can serve as the starting point for future research.

10.6.1 Experimental Setup

The signal path for the trials is shown in Figure 10.18. The setup in the laboratory can be seen in Figure 10.19 and consisted of an RF signal generator, the SMA cables, a balun, the evaluation board (ADC and FPGA), a PC (running client software) and a USB cable.

Figure 10.20 shows the sampled spectrum at Nyquist rate using the original device firmware and software provided by Texas Instruments. Figures in 10.21 show typical captures during the trials by the evaluation module. It was observed that the algorithm would only converge at a very small number of iterations, typically 1 to 4 and diverge soon after that. By lowering the value of the regulariser it is possible to remedy this situation slightly but at the expense of recovering, possibly noise components. The impact was a limited ability to reconstruct multiple emissions. Some background emissions would not be reported when the higher power control signal was transmitting.

To follow this, it was also observed that the value of the regulariser greatly affects the initial guess and subsequent steps. Even though such a situation does not arise in the theoretical analysis of the AMP algorithm [32] this can be possibly attributed to

Figure 10.19: *Setup in the laboratory.*

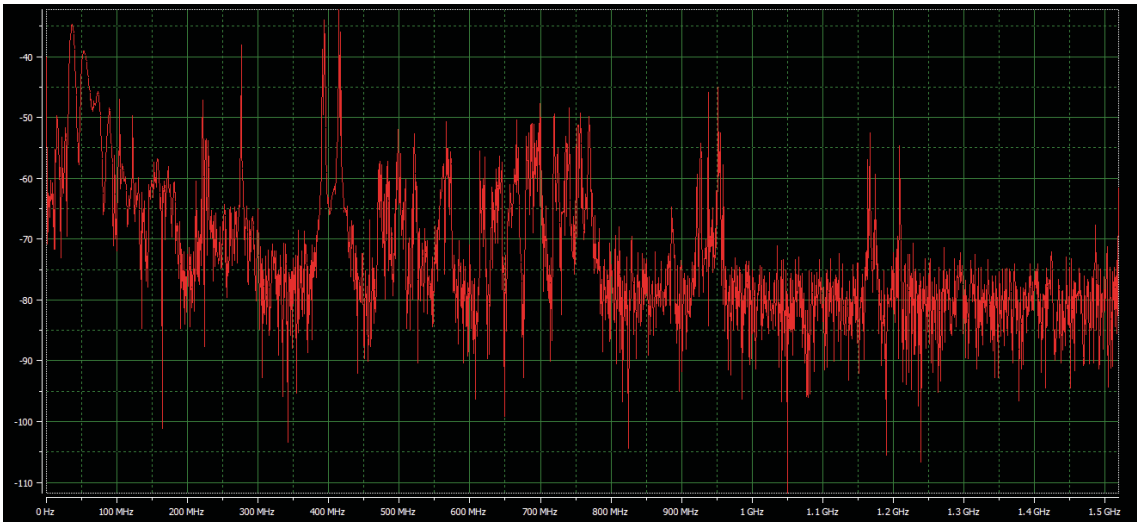
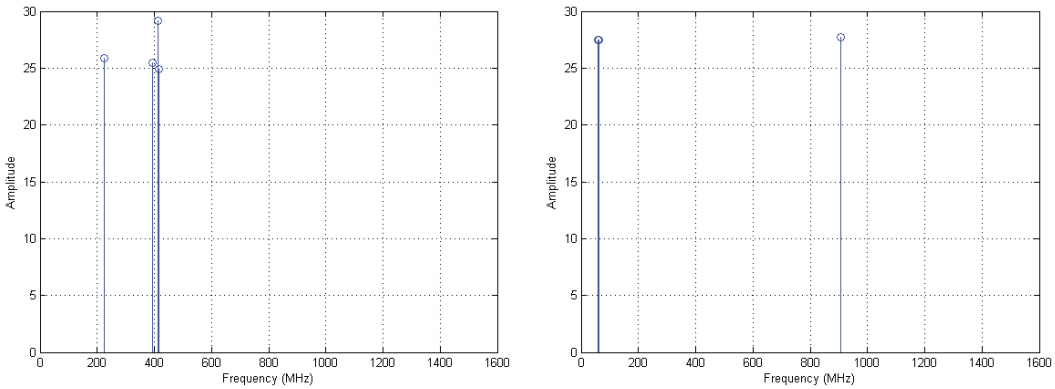


Figure 10.20: *Typical spectrum during the trials.*

the fixed point arithmetic and/or erratic FPGA behaviour as post-route simulations have not been carried out. The regulariser value should only affect the convergence speed.

The algorithm runs with two parameters to be predetermined, the maximum number of iterations and the regulariser value. In order to set these two values during the tests a trial and error procedure was followed by first increasing the value of the regulariser until a reasonable result was acquired and then increase the iteration count to fine tune the result.

The trial demonstrated that the prototype equipment was approximately 30dB's less sensitive than the commercial spectrum analyser used for comparison. This was expected because the spectrum analyser detects signals in a considerably narrower bandwidth (lower noise). However, it was easily demonstrated that the C.S equipment had a higher probability of capturing short duration transmissions than the



(a) **(b)**

Figure 10.21: *(a) Personal Radio Module transmission. (b) GSM transmission.*

spectrum analyser because of the latter's swept super-heterodyne architecture.

10.7 Conclusions

There are certain limitations linked to the chosen hardware that were proven impossible to overcome and a sub-optimal solution had to be followed. The lesson learnt from these limitations was that significant gains from sparse recovery methods can be possible in very specific scenarios and applications.

In the algorithmic part, a compressed sensing algorithm was implemented on an FPGA device. There are certain aspects of the algorithm that are not covered by the existing theory and limit to a great extent the performance. Careful and targeted research is possible to provide solutions to these problems. To be more specific, the performance of sparse recovery algorithms under limited numerical precision is rarely studied. Attempts have been made in order to understand these limitations in an empirical manner.

The platform was tested in laboratory conditions and field tests have also been carried out. The experience of working with such novel techniques in real-life was proven to be highly rewarding in every aspect.

Appendix A

Appendix

A.1 Matrix Identities

The Matrix Inversion Lemma

$$(\boldsymbol{A}+\boldsymbol{BCD})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{DA}^{-1}. \tag{A.1}$$

The Positive Definite Identity

If $\boldsymbol{A}$ and $\boldsymbol{C}$ are positive definite then the following holds,

$$\left(\boldsymbol{A}^{-1}+\boldsymbol{B}^{T}\boldsymbol{C}^{-1}\boldsymbol{B}\right)\boldsymbol{B}^{T}\boldsymbol{C}^{-1}=\boldsymbol{AB}^{T}\left(\boldsymbol{BAB}^{T}+\boldsymbol{C}\right)^{-1} \tag{A.2}$$

A.2 The Gaussian Distribution

The multivariate Gaussian distribution for a vector $\boldsymbol{x}\in\mathbb{R}^{m}$ is given by,

$$p(\boldsymbol{x})=\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{\sqrt{(2\pi)^{m}\cdot|\boldsymbol{\Sigma}|}}\cdot e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

where

$$\begin{aligned}\mathrm{E}[\boldsymbol{x}]&=\boldsymbol{\mu}\\ \mathrm{cov}[\boldsymbol{x}]&=\boldsymbol{\Sigma}.\end{aligned}$$

Given the marginal distribution for $\boldsymbol{x}$ and the conditional distribution for $\boldsymbol{y}$ given $\boldsymbol{x}$:

$$\begin{aligned}p(\boldsymbol{x})&=\mathcal{N}(\boldsymbol{\mu},\boldsymbol{A})\\ p(\boldsymbol{y}|\boldsymbol{x})&=\mathcal{N}(\boldsymbol{Hx},\boldsymbol{B})\end{aligned}$$

then the marginal distribution of $\boldsymbol{y}$ and the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ are given by:

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{H}\boldsymbol{\mu}, \boldsymbol{B}^{-1} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T\right) \tag{A.3}$$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\Sigma}(\boldsymbol{H}^T\boldsymbol{B}^{-1}\boldsymbol{y} + \boldsymbol{A}^{-1}\boldsymbol{\mu}), \boldsymbol{\Sigma}\right) \tag{A.4}$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{A}^{-1} + \boldsymbol{H}^T\boldsymbol{B}^{-1}\boldsymbol{H})^{-1}.$$

Bibliography

[1] M. R. Andersen, "Sparse Inference using Approximate Message Passing," M.Sc Thesis, Technical University of Denmark, 2014.

[2] D. Angelosante, G. Giannakis, and E. Grossi, "Compressed Sensing of Time-varying Signals," in *16th International Conference on Digital Signal Processing*. IEEE, 2009, pp. 1–8.

[3] S. Babacan, L. Mancera, R. Molina, and A. Katsaggelos, "Non-convex Priors in Bayesian Compressed Sensing," in *17th European Signal Processing Conference*, Aug 2009, pp. 110–114.

[4] S. Babacan, R. Molina, and A. Katsaggelos, "Bayesian Compressive Sensing Using Laplace Priors," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 53–63, 2010.

[5] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, "High-speed Compressed Sensing Reconstruction on FPGA Using OMP and AMP," in *19th IEEE International Conference on Electronics, Circuits and Systems*. IEEE, 2012, pp. 53–56.

[6] D. Baron, S. Sarvotham, and R. Baraniuk, "Bayesian Compressive Sensing via Belief Propagation," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 269–280, 2010.

[7] M. Bayati and A. Montanari, "The Dynamics of Message Passing on Dense Graphs, With Applications to Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 764–785, 2011.

[8] C. Berger, S. Zhou, J. Preisig, and P. Willett, "Sparse Channel Estimation for Multicarrier Underwater Acoustic Communication: From Subspace Methods to Compressed Sensing," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1708–1721, Mar 2010.

[9] C. Bishop, *Pattern Recognition and Machine Learning*. Springer New York, 2006, vol. 1.

[10] C. Bishop and M. Tipping, "Variational Relevance Vector Machines," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 46–53.

[11] T. Blumensath and M. Davies, "Iterative Hard Thresholding for Compressed Sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[12] T. Buchgraber, "Variational Sparse Bayesian Learning: Centralized and Distributed Processing," PhD Thesis, Graz University of Technology, Austria, 2013.

[13] E. Candes and J. Romberg, "Sparsity and Incoherence in Compressive Sampling," *Inverse Problems*, vol. 23, no. 3, p. 969, 2007.

[14] E. Candes and T. Tao, "The Dantzig Selector: Statistical Estimation When p is Much Larger Than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[15] E. Candes, "The Restricted Isometry Property and its Implications for Compressed Sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[16] E. Candes, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[17] E. Candes and T. Tao, "Decoding by Linear Programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[18] ——, "Near-optimal Signal Recovery from Random Projections: Universal Encoding Strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[19] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for Sparse Signal Recovery Using Kalman Filtering With Embedded Pseudo-measurement Norms and Quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, 2010.

[20] A. Carmi, P. Gurfil, D. Kanevsky, and B. Ramabhadran, "ABCS: Approximate Bayesian Compressed Sensing," *Human Language Technologies, IBM, Tech. Rep*, 2009.

[21] R. Carrillo, T. Aysal, and K. Barner, "Bayesian Compressed Sensing Using Generalized Cauchy Priors," in *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, 2010, pp. 4058–4061.

[22] A. Charles, M. Asif, J. Romberg, and C. Rozell, "Sparsity Penalties in Dynamical System Estimation," in *Conference on Information Sciences and Systems*. IEEE, 2011, pp. 1–6.

[23] C. Chen and M. Zoltowski, "Bayesian Sparse Channel Estimation and Tracking," in *IEEE Statistical Signal Processing Workshop*, 2012, pp. 472–475.

[24] Y. Chen, Y. Gu, and A. Hero, "Sparse LMS for System Identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3125–3128.

[25] L. D and Z. Wang, "DTMB-A Private Transmissions Test," Tsinghua Univercity.

[26] W. Dai and O. Milenkovic, "Subspace Pursuit for Compressive Sensing Signal Reconstruction," *IEEE Transactions on Information Theory*, vol. 55, pp. 2230–2249, 2009.

[27] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive Greedy Approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[28] D. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[29] D. Donoho and M. Elad, "Optimally Sparse Representation in General (nonorthogonal) Dictionaries via $\ell 1$ Minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[30] D. Donoho and X. Huo, "Uncertainty Principles and Ideal Atomic Decomposition," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[31] D. Donoho, A. Maleki, and A.Montanari, "The Noise-sensitivity Phase Transition in Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6920–6941, 2011.

[32] D. Donoho, A. Maleki, and A. Montanari, "Message-passing Algorithms for Compressed Sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[33] ——, "How to design message passing algorithms for compressed sensing," *preprint*, 2011.

[34] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[35] M. Duarte and M. Davenport, "Single-pixel Camera," 2008.

[36] M. Elad, *Sparse and Redundant Representations*. Springer, 2010.

[37] A. Faul and M. Tipping, "Analysis of Sparse Bayesian Learning," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 383–389.

[38] P. Feng and Y. Y. Bresler, "Spectrum-blind Minimum-rate Sampling and Reconstruction of Multiband Signals," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3. IEEE, 1996, pp. 1688–1691.

[39] J. Filos, E. Karseras, W. Dai, and S. Yan, "Tracking Dynamic Sparse Signals With Hierarchical Kalman Filters: A Case Study," in *18th International Conference on Digital Signal Processing*. IEEE, 2013, pp. 1–6.

[40] F.J. Beutler and O. AZ. Leneman, "The Spectral Analysis of Impulse Processes," *Information and Control*, vol. 12, no. 3, pp. 236–258, 1968.

[41] S. Foucart, "A Note on Guaranteed Sparse Recovery via ℓ1-minimization," *Applied and Computational Harmonic Analysis*, vol. 29, no. 1, pp. 97–103, 2010.

[42] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Bayesian Nonparametric Inference of Switching Dynamic Linear Models," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1569–1585, 2011.

[43] A. Gelman, A. Jakulin, M. Pittau, and Y. Su, "A Weakly Informative Default Prior Distribution for Logistic and Other Regression Models," *The Annals of Applied Statistics*, pp. 1360–1383, 2008.

[44] G. Golub and C. V. Loan, *Matrix Computations*. JHUP, 2012, vol. 3.

[45] I. Gorodnitsky and B. Rao, "Sparse Signal Reconstruction from Limited Data Using FOCUSS: A Re-weighted Minimum Norm Algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[46] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[47] K. Hwang and S. Choi, "Blind Equalization Method Based on Sparse Bayesian Learning," in *IEEE Vehicular Technology Conference*, 2008, pp. 658–662.

[48] S. Hwang and P. Schniter, "Efficient Multicarrier Communication for Highly Spread Underwater Acoustic Channels," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 1674–1683, Dec 2008.

[49] S. Ji, D. Dunson, and L. Carin, "Multitask Compressive Sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 92–106, 2009.

[50] J. Jin, Y. Gu, and S. Mei, "A Stochastic Gradient Approach on Compressive Sensing Signal Reconstruction Based on Adaptive Filtering Framework," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 409–420, 2010.

[51] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[52] E. Karseras and W. Dai, "A Fast Variational Approach for Bayesian Compressive Sensing with Informative Priors," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 5242–5246.

[53] E. Karseras, K. Leung, and W. Dai, "Bayesian Compressed Sensing: Improving Inference," in *IEEE China Summit International Conference on Signal and Information Processing*, Jul 2013, pp. 365–369.

[54] ——, "Hierarchical Bayesian Kalman Filters for Wireless Sensor Networks," in *21st European Signal Processing Conference*. IEEE, 2013, pp. 1–5.

[55] ——, "Tracking Dynamic Sparse Signals Using Hierarchical Bayesian Kalman Filters," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6546–6550.

[56] ——, "Tracking Dynamic Sparse Signals With Kalman Filters: Framework and Improved Inference," in *10th International Conference on Sampling Theory and Applications*, 2013, pp. 224–227.

[57] F. Li, J. Fang, H. Duan, Z. Chen, and H. Li, "Computationally Efficient Sparse Bayesian Learning via Generalized Approximate Message Passing," *arXiv:1501.04762*, 2015.

[58] J. Ling, X. Tan, T. Yardibi, J. Li, M. Nordenvaad, H. He, and K. Zhao, "On Bayesian Channel Estimation and FFT-Based Symbol Detection in MIMO Underwater Acoustic Communications," *IEEE Journal of Oceanic Engineering*, vol. 39, pp. 59–73, Jan 2014.

[59] W. Lu and N. Vaswani, "Modified Compressive Sensing for Real-time Dynamic MR Imaging," in *16th IEEE International Conference on Image Processing*, Nov 2009, pp. 3045–3048.

[60] ——, "Modified Basis Pursuit Denoising (modified-BPDN) for Noisy Compressive Sensing with Partially known Support," in *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, 2010, pp. 3926–3929.

[61] ——, "Exact Reconstruction Conditions for Regularized Modified Basis Pursuit," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2634–2640, 2012.

[62] ——, "Regularized modified BPDN for Noisy Sparse Reconstruction with Partial Erroneous Support and Signal Value Knowledge," *16th IEEE International Conference on Signal Processing*, vol. 60, no. 1, pp. 182–196, 2012.

[63] C. Luo and J. McClellan, "Discrete Random Sampling Theory," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 5430–5434.

[64] D. Madigan, J. York, and D. Allard, "Bayesian Graphical Models for Discrete Data," *International Statistical Review/Revue Internationale de Statistique*, pp. 215–232, 1995.

[65] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic press, 2008.

[66] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An Adaptive Greedy Algorithm With Application to Nonlinear Communications," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 2998–3007, 2010.

[67] M. Mishali and Y. Eldar, "Reduce and Boost: Recovering Arbitrary Sets of Jointly Sparse Vectors," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4692–4702, 2008.

[68] M. .Mishali and Y. Eldar, "From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, 2010.

[69] A. Montanari, "Graphical Models Concepts in Compressed Sensing," *Compressed Sensing: Theory and Applications*, pp. 394–438, 2012.

[70] S. Mukherjee and T. Speed, "Network Inference Using Informative Priors," *Proceedings of the National Academy of Sciences*, vol. 105, no. 38, pp. 14 313–14 318, 2008.

[71] NASA, "Space-based Measurements of Ozone and Air quality," http://ozoneaq.gsfc.nasa.gov/, 2012, [Online; accessed 19-July-2008].

[72] D. Needell and J. Tropp, "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[73] D. Needell and R. Vershynin, "Signal Recovery From Incomplete and Inaccurate Measurements via Regularized Orthogonal Matching Pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, 2010.

[74] N. Pedersen, C. Manchon, D. Shutin, and B. Fleury, "Application of Bayesian Hierarchical Prior Modeling to Sparse Channel Estimation," in *IEEE International Conference on Communications*, 2012, pp. 3487–3492.

[75] K. Petersen, M. Pedersen, M. Syskind *et al.*, "The Matrix Cookbook," *Technical University of Denmark*, 2012.

[76] R. Prasad and C. Murthy, "Bayesian Learning for Joint Sparse OFDM Channel Estimation and Data Detection," in *IEEE Global Telecommunications Conference*, 2010, pp. 1–6.

[77] C. Qiu and N. Vaswani, " Compressive Sensing on the Least Squares and Kalman Filtering Residual for Real-time Dynamic MRI and Video Reconstruction," in *IEEE Transactions on Image Processing*, 2009.

[78] N. Radford and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*. Springer, 1998, pp. 355–368.

[79] M. Raghavendra and K. Giridhar, "Improving Channel Estimation in OFDM Systems for Sparse Multipath Channels," *IEEE Signal Processing Letters*, vol. 12, pp. 52–55, Jan 2005.

[80] T. Ragheb, J. Laska, H. Nejati, S. Kirolos, R. Baraniuk, and Y. Massoud, "A Prototype Hardware for Random Demodulation Based Compressive Analog-to-digital Conversion," in *51st Midwest Symposium on Circuits and Systems*, Aug 2008, pp. 37–40.

[81] S. Rangan, "Generalized Approximate Message Passing for Estimation With Random Linear Mixing," in *IEEE International Symposium on Information Theory Proceedings*. IEEE, 2011, pp. 2168–2172.

[82] B. Rao and D. Wipf, "Comparing the Effects of Different Weight Distributions on Finding Sparse Representations," in *Advances in Neural Information Processing Systems*, 2005, pp. 1521–1528.

[83] C. Shannon, "Communication In The Presence Of Noise," *Proceedings of the IEEE*, vol. 86, no. 2, pp. 447–457, 1998.

[84] J. Shihao, X. Y. Xue, and L. Carin, "Bayesian Compressive Sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346 –2356, 2008.

[85] D. Shutin, T. Buchgraber, S. Kulkarni, and H. Poor, "Fast Adaptive Variational Sparse Bayesian Learning with Automatic Relevance Determination," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2011, pp. 2180–2183.

[86] ——, "Fast Variational Sparse Bayesian Learning With Automatic Relevance Determination for Superimposed Signals," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6257–6261, 2011.

[87] G. Su, J. Jin, Y. Gu, and J. Wang, "Performance Analysis of l_0Norm Constraint Least Mean Square Algorithm," *arXiv preprint arXiv:1203.1535*, 2012.

[88] X. Tan and J. Li, "Computationally Efficient Sparse Bayesian Learning via Belief Propagation," *IEEE Transactions on Signal Processing,*, vol. 58, no. 4, pp. 2010–2021, Apr 2010.

[89] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *The Journal of Machine Learning Eesearch*, vol. 1, pp. 211–244, 2001.

[90] M. Tipping and A. Faul, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models," in *International Workshop on Artificial Intelligence and Statistics*, 2003.

[91] J. Tropp, "Greed is good: Algorithmic Results for Sparse Approximation," *IEEE Transactions of Information Theory*, vol. 50, pp. 2231 – 2242, Oct 2004.

[92] J. Tropp, J. Laska, M. Duarte, J. Romberg, and R. Baraniuk, "Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, 2010.

[93] J. Tropp, M. Wakin, M. Duarte, D. Baron, and R. Baraniuk, "Random Filters for Compressive Sampling and Reconstruction," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3. IEEE, 2006, pp. III–III.

[94] D. Tzikas, C. Likas, and N. Galatsanos, "Sparse Bayesian Modeling with Adaptive Kernel Learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 926–937, 2009.

[95] P. Vaidyanathan and P. Pal, "Sparse Sensing With Co-prime Samplers and Arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, 2011.

[96] J.-J. van de Beek, O. Edfors, M. Sandell, S. Wilson, and P. O. Borjesson, "On Channel Estimation in OFDM Systems," in *IEEE Vehicular Technology Conference*, 1995, pp. 815–819.

[97] N. Vaswani, "Kalman Filtered Compressed Sensing," in *15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 893–896.

[98] ——, " KF-CS: Compressive Sensing on Kalman Filtered Residual," *arXiv:0912.1628*, 2009.

[99] ——, "Analyzing Least Squares and Kalman Filtered Compressed Sensing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr 2009, pp. 3013–3016.

[100] ——, " LS-CS-Residual (LS-CS): Compressive Sensing on Least Squares Residual," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4108–4120, Aug 2010.

[101] ——, "Stability (over time) of Modified-CS for Recursive Causal Sparse Reconstruction," in *48th Annual Allerton Conference on Communication, Control and Computing*. IEEE, 2010, pp. 1722–1729.

[102] N. Vaswani and W. Lu, "Modified-CS: Modifying Compressive Sensing for Problems with Partially known Support," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4595–4607, 2010.

[103] D. Wipf, J. Palmer, B. Rao, and K. Kreutz-Delgado, "Performance Evaluation of Latent Variable Models with Sparse Priors," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, Apr 2007, pp. II–453 –II–456.

[104] D. Wipf, *Bayesian Methods for Finding Sparse Representations*. ProQuest, 2006.

[105] D. Wipf and B. Rao, "Bayesian Learning for Sparse Signal Reconstruction," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6. IEEE, 2003, pp. VI–601.

[106] ——, " *ell*_0-norm Minimization for Basis Selection," in *Advances in Neural Information Processing Systems*, 2004, pp. 1513–1520.

[107] ——, "Sparse Bayesian Learning for Basis Selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153 – 2164, Aug 2004.

[108] D. Wipf, B. Rao, and S. Nagarajan, "Latent Variable Bayesian Models for Promoting Sparsity," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6236–6255, 2011.

[109] D. Wipf and N. Srikantan, "A new view of automatic relevance determination," in *Advances in Neural Information Processing Systems*, 2008, pp. 1625–1632.

[110] Z. Yang, L. Xie, and C. Zhang, "Bayesian Compressed Sensing With New Sparsity-inducing Prior," *arXiv:1208.6464*, 2012.

[111] Z. Zhang, "Comparison of Sparse Signal Recovery Algorithms with Highly Coherent Dictionary Matrices: The Advantage of T-MSBL," *Research Note*, 2012.

[112] J. Ziniel, L. Potter, and R. Schniter, "Tracking and Smoothing of Time-varying Sparse Signals via Approximate Belief Propagation," in *Asilomar Conference on Signals, Systems and Computers*. IEEE, 2010, pp. 808–812.