ABSTRACT


Parametric signal models are used in a multitude of signal processing applications. This thesis deals with signals for which there are many candidate models, and it is not a priori known which model is the most appropriate one. The first part of the thesis treats cases for which the set of models is relatively small, so that it is possible to evaluate each model in the set separately. The second part deals with sparse models, i.e., models sharing the same parameter vector, but for which any combination of zero valued and non-zero valued parameters is possible. Sparse models appear in a variety of applications, such as statistical data analysis, communications, and active sensing, such as radar and non-destructive testing.

An important problem considered in the two parts of the thesis is that of model selection, i.e., how to select the most appropriate model (according to some criterion) from the set of candidates. To this end, both the classical information criterion (IC) approaches, such as AIC, BIC and GIC, as well as maximum a posteriori probability based methods derived in the Bayesian framework are used.

Another problem which is treated in this thesis is that of parameter estimation, i.e., how to most accurately estimate the parameters present in a set of candidate models. Specifically, the multi-modeling, or model averaging, approach is studied. In such an approach the estimated parameters from several candidate models are weighted together. Again, both the IC framework and the Bayesian framework are used to solve this problem.

Finally, the task of symbol detection for applications in communications is considered. The maximum a posteriori probability symbol detector is derived for a few different channel models.

Keywords: model selection; model order selection; model averaging; nested models; sparse models; Bayesian inference; MMSE estimation; MAP estimation; ML estimation; AIC; BIC; GIC; RAKE receivers; pulse compression; radar; linear models; linear regression models

Yngve Selén, Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden.
Acknowledgments

During the work which eventually led to this thesis I have been helped by a great number of great people.

The first person I would like to thank is my supervisor, Prof. Petre Stoica. Not only has he been a fabulous teacher, but he has also taken care to create a happy and creative working environment. Also, his ability to find interesting problems can not be understated. Indeed, without his guidance, this thesis would never have been written. He has my most sincere appreciation.

I would also like to thank all my co-authors, especially Dr. Erik G. Larsson, since this fall at Linköping University, who has had a mentor-like role during the last years of my PhD studies. I have learned a lot from him. Special thanks also goes to “Petre’s Pack”—current and former students of Prof. Stoica—especially to Nic-E-Rich: to Niclas for co-authorship and great travel company, to Erik for interesting discussions and enthusiasm, and to Richard for matrix mastership and for always having time for questions. Thanks also to Andreas for conference gourmetification.

During my years as a PhD student I have had the privilege of traveling to many exotic places. I would like to thank the following funding agencies for their support: Anna Maria Lundins stipendiefond at Smålands nation in Uppsala, C. F. Liljewalchs resestipendiefond, Rektors resebidrag från Wallenbergstiftelsen and Noréus stipendiistiftelse, all at Uppsala university, the ICASSP 2007 travel grant, and the Royal Swedish Academy of Sciences.

I am grateful to Prof. Jian Li at the University of Florida, Gainesville, USA, for co-authorship and for hosting me during six sunny months in 2003. Also, I’d like to thank Prof. Sabine Van Huffel at Katholieke Universiteit Leuven, Belgium, for cooperation in the field of magnetic resonance spectroscopy.

The working atmosphere during my years at the Department of Information Technology has been most pleasant. I would like to thank all my colleagues—professors, students, and administrative personnel alike—for this. Thank you for all coffee breaks, wine tastings and (sometimes very) weird discussions (thanks also goes to Wikipedia for giving (most) answers). Hats off to creators of chocolate, makers of music, and brewers of beer, for adding spice and flair to life. ...and blessed be the cheese makers!

My friends and family have always had, and continue to have, an important
and very positive effect on my life. Thanks to all of you for being wonderful!

Finally, I would like to thank my little solstråle Stina, for happiness and love, and for making everything better!
# Contents

Acknowledgments 3  
Modellselektion och gles modellering (Summary in Swedish) 11  
Glossary and Notation 17  

1 Introduction 21  
1.1 Why is the Topic of this Thesis Important? 21  
1.2 Signal Processing 21  
1.3 Parametric Models in Signal Processing 22  
1.3.1 Model Characterization 24  
1.3.2 Model Selection 25  
1.3.3 Nested Models 28  
1.3.4 Sparse Models 29  
1.3.5 The Use of a Model 31  
1.4 Bayesian Inference 31  
1.5 The Gaussian Distribution 33  
1.5.1 The Central Limit Theorem 34  
1.5.2 The Maximum Entropy Principle 34  
1.5.3 Computational Convenience 34  
1.6 Thesis Outline 34  
1.7 Beside the Thesis 39  
1.8 Beyond the thesis 40  

I Non-Sparse Modeling 41  

2 A Review of the AIC, GIC and BIC Rules 43  
2.1 Introduction 43  
2.2 Maximum Likelihood Parameter Estimation 44  
2.3 Useful Mathematical Preliminaries and Outlook 47  
2.3.1 Maximum \textit{a Posteriori} Selection Rule 47  
2.3.2 Kullback-Leibler Information 50
### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3</td>
<td>Outlook: Theoretical and Practical Perspectives</td>
<td>51</td>
</tr>
<tr>
<td>2.4</td>
<td>Direct Kullback-Leibler (KL) Approach: No-Name Rule</td>
<td>51</td>
</tr>
<tr>
<td>2.5</td>
<td>Cross-Validatory KL Approach: The AIC Rule</td>
<td>53</td>
</tr>
<tr>
<td>2.6</td>
<td>Generalized Cross-Validatory KL Approach: The GIC Rule</td>
<td>56</td>
</tr>
<tr>
<td>2.7</td>
<td>Bayesian Approach: The BIC Rule</td>
<td>58</td>
</tr>
<tr>
<td>2.8</td>
<td>Summary</td>
<td>61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>Cross-Validation Rules for Model Order Selection</th>
<th>65</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction and General Discussion</td>
<td>65</td>
</tr>
<tr>
<td>3.2</td>
<td>Short-Term and Long-Term Validation Rules</td>
<td>66</td>
</tr>
<tr>
<td>3.3</td>
<td>Numerical Examples</td>
<td>70</td>
</tr>
<tr>
<td>3.4</td>
<td>Concluding Remarks</td>
<td>72</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>On Information Criteria and the Generalized Likelihood Ratio Test of Model Selection</th>
<th>83</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction: The IC and the GLRT</td>
<td>84</td>
</tr>
<tr>
<td>4.2</td>
<td>The Main Result: The Equivalence Between GIC and GLRT</td>
<td>85</td>
</tr>
<tr>
<td>4.3</td>
<td>A Corollary: Using the GIC Rule for Sparse Models</td>
<td>86</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Numerical Example</td>
<td>87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th>Multi-Model Approach to Model Selection</th>
<th>91</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction and Preliminaries</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>AIC and BIC Rules for Single-Model Selection</td>
<td>93</td>
</tr>
<tr>
<td>5.3</td>
<td>AIC and BIC Rules for Multi-Model Selection</td>
<td>94</td>
</tr>
<tr>
<td>5.4</td>
<td>Numerical Study of Performance</td>
<td>97</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Performance Measures</td>
<td>97</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Numerical Examples</td>
<td>99</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusions</td>
<td>108</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>Adaptive Equalization for Frequency-Selective Channels of Unknown Length</th>
<th>109</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>109</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Problem Formulation and Background</td>
<td>109</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Previous and Related Work</td>
<td>111</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Contributions</td>
<td>111</td>
</tr>
<tr>
<td>6.2</td>
<td>Preliminaries</td>
<td>112</td>
</tr>
<tr>
<td>6.3</td>
<td>Gaussian Mixture Channel Model</td>
<td>113</td>
</tr>
<tr>
<td>6.3.1</td>
<td>The Optimal Receiver for Known ${\mu_n, \Lambda_n, P(M_n), \sigma^2}$</td>
<td>114</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Discussion of the Optimal Receiver</td>
<td>115</td>
</tr>
<tr>
<td>6.4</td>
<td>Training-Based Estimation of the Mixture Model Parameters</td>
<td>115</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Estimation of ${\mu_n, \Lambda_n, \sigma^2}$ from a Training Sequence</td>
<td>116</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Estimation of $P(M_n)$</td>
<td>117</td>
</tr>
<tr>
<td>6.5</td>
<td>The Trained Multi-Model Detector and Its Implementation</td>
<td>118</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Trained Multi-Model Detector</td>
<td>118</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Possible Approximations of the Trained Multi-Model Detector (6.13)+(6.27)</td>
<td>119</td>
</tr>
<tr>
<td>6.6</td>
<td>Numerical Examples</td>
<td>120</td>
</tr>
<tr>
<td>6.7</td>
<td>Concluding Remarks</td>
<td>121</td>
</tr>
</tbody>
</table>
### 7 Empirical Bayes Linear Regression with Unknown Model Order

7.1 Introduction .................................................. 125
  7.1.1 Problem Formulation ................................. 125
  7.1.2 Related Work ......................................... 126
  7.1.3 Contributions of This Work ......................... 126
7.2 Bayesian Model Order Selection and Parameter Estimation ........ 127
  7.2.1 Optimal Model Order Selection .................... 127
  7.2.2 The MMSE Estimate of \( h \) ....................... 128
  7.2.3 A Note on Complexity ................................ 129
7.3 Empirical Bayesian Estimation ................................ 129
  7.3.1 Estimation of \( \{ \gamma_j^2 \}_{j=1}^n \) by Parameterization .... 131
  7.3.2 Estimation of \( \{ \gamma_j^2 \}_{j=1}^n \) by Penalization ...... 132
  7.3.3 Multiple Coefficient Vector Realizations ............ 133
7.4 Numerical Examples ............................................. 134
  7.4.1 Example 1: Uncorrelated Regressors ............... 135
  7.4.2 Example 2: Correlated Regressors .................. 137
  7.4.3 Example 3: Variance Profile Mismatch .............. 140
  7.4.4 Example 4: Multiple Coefficient Vector Realizations . 141
7.5 Conclusions .................................................. 141

### II Sparse Modeling

8 A Model Averaging Approach for Sparse Communication Channel Equalization ........ 145
  8.1 Introduction .............................................. 145
  8.2 Sparse Channel Averaging .............................. 146
    8.2.1 Information Criteria for Selecting a Single Channel ... 148
    8.2.2 Information Criteria for Channel Averaging ........ 148
    8.2.3 Information Criteria for Sparse Channel Averaging ... 149
  8.3 Numerical Example ........................................ 150
  8.3.1 Channel Model ....................................... 150
  8.4 Conclusions ................................................ 152
9 Linear Regression With a Sparse Parameter Vector .................... 155
  9.1 Introduction .............................................. 155
  9.1.1 Problem Formulation ................................ 155
  9.1.2 Examples of Applications for Sparse Models ......... 156
  9.1.3 Related Work on Regression with Sparse Models ...... 156
  9.1.4 Contribution of This Work .......................... 159
  9.2 Model ...................................................... 160
  9.3 MMSE Parameter Estimation .............................. 161
    9.3.1 Derivation of the Estimate ....................... 161
    9.3.2 Algorithm for Computing the Approximate MMSE Estimate . 163
    9.3.3 Computational Complexity ....................... 164
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3.4</td>
<td>Implementation Considerations</td>
<td>164</td>
</tr>
<tr>
<td>9.4</td>
<td>Model Selection</td>
<td>164</td>
</tr>
<tr>
<td>9.5</td>
<td>Strategies for Handling Unknown {P(H_i), \gamma^2, \sigma^2}</td>
<td>165</td>
</tr>
<tr>
<td>9.5.1</td>
<td>Bayesian Estimator with known <em>a Priori</em> Distributions for the Parameters</td>
<td>165</td>
</tr>
<tr>
<td>9.5.2</td>
<td>Empirical Bayesian Version of the Estimator</td>
<td>166</td>
</tr>
<tr>
<td>9.6</td>
<td>Numerical Examples</td>
<td>167</td>
</tr>
<tr>
<td>9.6.1</td>
<td>Example 1: Estimation Performance Comparison with Uncorrelated Regressors</td>
<td>168</td>
</tr>
<tr>
<td>9.6.2</td>
<td>Example 2: Analysis of the Sensitivity to the <em>a Priori</em> Parameters</td>
<td>169</td>
</tr>
<tr>
<td>9.6.3</td>
<td>Example 3: Estimation Performance Comparison with Correlated Regressors</td>
<td>170</td>
</tr>
<tr>
<td>9.6.4</td>
<td>Example 4: Model Selection</td>
<td>170</td>
</tr>
<tr>
<td>9.7</td>
<td>Concluding Remarks</td>
<td>171</td>
</tr>
<tr>
<td>10</td>
<td>RAKE Receiver for Channels with a Sparse Impulse Response</td>
<td>179</td>
</tr>
<tr>
<td>10.1</td>
<td>Introduction</td>
<td>179</td>
</tr>
<tr>
<td>10.2</td>
<td>Problem Formulation</td>
<td>180</td>
</tr>
<tr>
<td>10.3</td>
<td>The Optimal Receiver</td>
<td>181</td>
</tr>
<tr>
<td>10.4</td>
<td>An Empirical Bayesian Receiver</td>
<td>184</td>
</tr>
<tr>
<td>10.5</td>
<td>Numerical Examples</td>
<td>185</td>
</tr>
<tr>
<td>10.5.1</td>
<td>Sparse Toy Channel Model</td>
<td>185</td>
</tr>
<tr>
<td>10.5.2</td>
<td>Non-Sparse Toy Channel Model—Robustness to Mismatch</td>
<td>186</td>
</tr>
<tr>
<td>10.5.3</td>
<td>Indoor Radio Channel Model</td>
<td>188</td>
</tr>
<tr>
<td>10.6</td>
<td>Concluding Remarks</td>
<td>191</td>
</tr>
<tr>
<td>11</td>
<td>Estimation of Semi-Sparse Radar Range Profiles</td>
<td>193</td>
</tr>
<tr>
<td>11.1</td>
<td>Introduction</td>
<td>193</td>
</tr>
<tr>
<td>11.2</td>
<td>Signal Model and Previous Work</td>
<td>195</td>
</tr>
<tr>
<td>11.2.1</td>
<td>Matched Filter Estimate</td>
<td>195</td>
</tr>
<tr>
<td>11.2.2</td>
<td>Least Squares Estimate</td>
<td>196</td>
</tr>
<tr>
<td>11.2.3</td>
<td>Non-Sparse MMSE Estimate</td>
<td>198</td>
</tr>
<tr>
<td>11.2.4</td>
<td>Reiterative MMSE Approaches</td>
<td>198</td>
</tr>
<tr>
<td>11.3</td>
<td>Semi-Sparse MMSE Estimate</td>
<td>199</td>
</tr>
<tr>
<td>11.3.1</td>
<td>Algorithm for Finding (\hat{x}_{\text{MMSE}})</td>
<td>201</td>
</tr>
<tr>
<td>11.3.2</td>
<td>Target Detection</td>
<td>202</td>
</tr>
<tr>
<td>11.3.3</td>
<td>Computational Complexity</td>
<td>203</td>
</tr>
<tr>
<td>11.3.4</td>
<td>The Model Versus Reality Issue</td>
<td>204</td>
</tr>
<tr>
<td>11.4</td>
<td>Numerical Examples</td>
<td>204</td>
</tr>
<tr>
<td>11.4.1</td>
<td>Example 1: Estimation and Detection with Closely Spaced Targets</td>
<td>205</td>
</tr>
<tr>
<td>11.4.2</td>
<td>Example 2: Estimation with Target Signature</td>
<td>209</td>
</tr>
<tr>
<td>11.4.3</td>
<td>Example 3: Estimation With <em>a Priori</em> Parameter Mismatch</td>
<td>209</td>
</tr>
<tr>
<td>11.4.4</td>
<td>Example 4: Average Computation Time</td>
<td>215</td>
</tr>
<tr>
<td>11.5</td>
<td>Concluding Remarks</td>
<td>217</td>
</tr>
</tbody>
</table>
## Appendices

A. Cyclic Minimizers, Majorization Techniques, and the Expectation-Maximization Algorithm

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Introduction</td>
<td>221</td>
</tr>
<tr>
<td>A.2 Cyclic Minimizer</td>
<td>222</td>
</tr>
<tr>
<td>A.3 Majorization Technique</td>
<td>223</td>
</tr>
<tr>
<td>A.4 EM Algorithm</td>
<td>224</td>
</tr>
</tbody>
</table>

B. Proof of (3.22) (Chapter 3)                | 227  |

C. Minimum Error Probability Threshold (Chapter 11) | 229  |

Bibliography                                   | 230  |
Modellselektion och gles modellering  
(Summary in Swedish)

Här följer en kort sammanfattning, på svenska, av avhandlingens innehåll. Avhandlingen sorteras under ämnet *signalbehandling*. Först ges en kortfattad introduktion till ämnet, som sedan följs av en redogörelse av avhandlingens egentliga innehåll.

Varför är detta viktigt?

Utvecklingen inom signalbehandling och andra relaterade fält har haft en enorm inverkan på den moderna människans sätt att leva. Vi använder signalbehandling för att kommunicera med mobiltelefoner eller över Internet, för att koda vår musik i mp3-format så att vi kan bära den med oss i portabla spelare, för att tolka medicinskt data vid sjukdomsdiagnostik, för att avläsa vår geografiska position med hjälp av GPS, etc. (se tabell 1 för ytterligare några exempel). Många signalbehandlingsmetoder är baserade på så kallade parametriska signalmodeller, som behandlas i denna avhandling. Nya resultat inom parametrisk modellering har därför stora möjligheter att påverka en rad tillämpningsområden. Några möjliga exempel är snabbare dataöverföring och lägre energiförbrukning för mobiltelefoner, mer tillförlitliga medicinska diagnoser, bättre väderprognoser, samt mer tillförlitliga jordbävnings- och tsunamivarningar.

Signalbehandling

I forskningsområdet signalbehandling sysslar man med representation, tolkning, manipulation och transformering av *signaler*. Med en signal avses vanligtvis en endimensionell sekwens över tiden, även om andra typer av signaler förekommer. I tabell 1 anges några exempel på signaler, samt möjliga signalbehandlingstillämpningar för dessa.

I denna avhandling diskuteras endast digital signalbehandling. Det innebär att de vanligtvis ursprungligen kontinuerliga signalerna har samplats (avlästs)
Tabell 1: Några exempel på signaler och möjliga tillämpningar för signalbehandling.

<table>
<thead>
<tr>
<th>Exempel på signal</th>
<th>Möjliga tillämpningar</th>
</tr>
</thead>
<tbody>
<tr>
<td>mätning av utomhustemperatur</td>
<td>finna periodiciteter i temperaturens förändringar, eller skatta temperaturens värde en timme in i framtiden</td>
</tr>
<tr>
<td>avläsningen av en seismometer</td>
<td>prediktera jordbävningar</td>
</tr>
<tr>
<td>det inspelade ljudet från en gitarrsträng</td>
<td>avgöra strängens ton i en stämpapparat</td>
</tr>
<tr>
<td>en patientens elektrokardiogram</td>
<td>bestämma patientens puls, eller avgöra om patienten lider av något hjärtproblem</td>
</tr>
<tr>
<td>inducerad spänning i en mobiltelefons antenn</td>
<td>transformera den emotagna signalen till mänskligt tal</td>
</tr>
</tbody>
</table>

i ett antal punkter, och att de avlästa värdena lagras i digital form. En signal betraktas alltså som en serie av siffror, där varje siffra motsvarar en position i den domän (till exempel tiden) från vilken signalen samplats.

En mer omfattande svensk generell introduktion till ämnet signalbehandling ligger utanför ramen för denna avhandling. För den intresserade rekommenderas istället [56; 58; 154].

**Parametriska modeller för signalbehandling**


Om vi låter $y(t)$ beskriva en signal samplad vid tidpunkterna $t = 0, \ldots, N-1$, kan vi modellera $y(t)$ med följande ekvation:

$$ y(t) = f(\theta, t) + \epsilon(t). \quad (1) $$

Ovan står $f(\cdot)$ för den parametriska modellfunktionen, $\theta$ för parametervektorn, och $\epsilon(t)$ är en term som beskriver skillnaden mellan modellfunktionen och den uppmätta signalen $y(t)$. Några vanliga modellfunktioner och deras respektive parametrar finns angivna i tabell 1.2 i den engelska introduktionen. Termen $\epsilon(t)$

1I allmänhet låter vi tidsindexet $t$ stå för sampelnumret, och inte den faktiska tid (i sekunder) vid vilken signalen samplades.
kallas vanligen brus\textsuperscript{2}, eller residual. I praktiska signalbehandlingstillämpningar förekommer alltid brus, eftersom ingen mätning kan vara perfekt. Hanteringen av detta brus är en mycket viktig del av ämnet signalbehandling.

Användbarheten av (1) är starkt beroende på hur väl $f(\theta, t)$ och de antaganden som gjorts beträffande $e(t)$ förnär beskriver signalen $y(t)$. År modellen dåligt vald så kommer resultatena baserade på (1) i bästa fall vara av lite värde, och i värsta fall vara direkt vilseledande. \textit{Modellseluktion} är konsten att välja det mest lämpliga uttrycket för $f(\theta, t)$ (och den mest lämpliga beskrivningen av $e(t)$) från en mängd olika kandidater. Detta beskrivs mer utförligt nedan.

\textbf{Modellseluktion}

För vissa tillämpningar är en lämplig modelfunktion $f(\theta, t)$ given. Till exempel vore det svårt att tänka sig att använda några andra ekvationer än Newtons lagar för att beskriva makroskopiska kroppars rörelser under påverkan av kända krafter (vid icke-relativistiska hastigheter). I många tillämpningar, däremot, är det endast känt vilken eller vilka typ av modellfunktioner som kan vara lämpliga att använda, och den exakta formen av modellfunktionen är okänd. I dessa fall finns det ett flertal modeller att välja mellan. Målet för modellseluktionen är att bestämma vilken av dessa modeller som är lämpligast för att beskriva $y(t)$.

\textbf{Exempel: Magnetresonansspektroskopi.} Det är välkänt, från underliggande fysikaliska och kemiska förhållanden, att magnetresonansspektroskopi\textsuperscript{1}data kan beskrivas väl som en summa av exponentiellt dämpade komplexa sinusvågor i additivt brus [124]. Däremot är det lämpliga antalet sinusvågor i allmänhet okänt. Det innebär att vi har ett antal, säg $K$, möjliga modellfunktioner, \( \{f_k(\theta, t)\}_{k=1}^K \), där varje modellfunktion beskriver en summa av ett unikt antal exponentiellt dämpade sinusvågor.

Vid första påseende verkar det frestande att välja en modell med hög komplexitet (i betydelsen många parametrar) för att beskriva $y(t)$. Det skulle ju resultera i en modell som sannolikt är flexibel nog för att kunna beskriva $y(t)$ väl. Ju mer flexibel en modell blir, desto bättre kommer nämligen modellen kunna anpassas efter det uppmätta datat (åtminstone då modellerna är nästlade; se nedan).

Det finns (åtminstone) två uppenbara problem med ovanstående resonemang. Till att börja med så noterar vi att ju flera parametrar vi har i vår modell, desto mer data krävs i allmänhet för att kunna skatta parametrarnas värden väl. Därför kan vi, för en signal med ett fikt antal datasampel, i allmänhet inte välja en signalmodell som inkluderar alla parametrar som kan tänkas inverka på signalen. Väljer vi ”för många” parametrar kommer parameterningsställningarna påverkas i hög grad av bruset, och därmed troligen avvika stort från sina ”sanna” (eller ”lämpliga”) värden. Ett annat problem med att

\textsuperscript{2}Notera att brusets egenskaper också vanligtvis beskrivs av ett antal parametrar, såsom medelvärde och varians. För att förenkla notationen skrivs inte dessa parametrar ut här.
välja en modell med ”för många” parametrar har med tolkningen av modellen att göra: Om valet av modell är det shutgiltiga målet med modellselukonen vill vi i allmänhet exkludera parametrar som sannolikt inte har inverkat på signalen.

Det första av de ovanstående två problemen är ett exempel på den synbarligen ständigt återkommande kompromissen mellan väntevärdesriktighet och varians [147]. För en modell med hög komplexitet kommer kvoten mellan antalet datasampel och antalet parametrar att få ett lågt värde. Detta leder till att variansen hos parameterskattningarna blir hög, även om de kan vara väntevärdesriktiga. För en modell med låg komplexitet, med ett högt värde för kvoten mellan antalet datasampel och antalet parametrar, kan däremot viktig parametern saknas. Detta kan leda till att de kvarvarande parametrarnas estimat inte blir väntevärdesriktiga, men i gendöld får låg varian.

Occams rakkniv beskriver en sund princip som spelar en viktig roll inom ämnet modellselukon. Denna princip säger i stort sett att av flera möjliga, i andra avseenden likvärdiga, förklaringar bör man välja den enklaste. För modellselukon kan detta översättas med om flera modeller kan beskriva datat med likvärdig precision bör den enklaste modellen väljas. Intressant är att notera att de vanligaste modellselukongalgoritmerna fungerar på precis detta sätt. Till exempel finns det, för de informationskriteriebaserade modellselukongalgoritmer som redogörs för i kapitel 2, en term som straffar komplexa modeller. Även inom Bayesiansk modellselukon finns det en motsvarande Occamfaktor [92].

Översikt

Här följer en kort översikt av innehållet i de olika delarna av avhandlingen.

Del I: Icke-gles modellering

Denna del, bestående av kapitel 2-7, behandlar huvudsakligen nästlade modeller. Mer specifikt behandlar denna del fall då mängden möjliga modeller är förhållandevis liten.3 En mängd modeller är nästlade om varje modell i mängden är ett specialfall av de mer komplexa modellerna i mängden. Vanliga exempel på nästlade modellmängder är polynom med olika gradtal (ett polynom med grad m kan beskrivas som ett polynom med grad n > m där alla koefficienter som motsvarar grader > m är nollvärda), autoregressiva (AR) modeller med olika ordning, finita impulsrespons- (FIR-) filter, etc. (se tabell 1.2 i den engelska introduktionen). För dessa fall kan komplexiteten hos en modell beskrivas med en heltalsvärd parameter: den så kallade modellordningen.

I kapitel 2 ges en översikt över några klassiska metoder för skattning av modellordningen; de så kallade informationskriterierna (IC). Dessa metoder

---

3I princip kan resultaten och metoderna i denna del användas för godtyckliga mängder av modeller; även för glesa modeller. Dock växer antalet möjliga glesa modeller exponentiellt med antalet möjliga parametrar (se diskussionen nedan). Direkt tillämpning av metoderna som beskrivs här är därför inte möjlig, eftersom kraven på beräkningskapacitet blir för stora.
jämförs sedan, i kapitel 3, med två tidigare introducerade korsvalideringsbaserade metoder. Dessa korsvalideringsbaserade metoder har fördelen att de ansätter färre antaganden om den signalgenererande processen än IC. De kan därför, i någon mening, sägas vara mer generellt tillämpbara. I kapitel 4 diskuteras en ekvivalens mellan det så kallade generaliserade likelihood-ratio-testet (GLRT) och IC. Av denna diskussion följer en metod som använder det lokala beteendet hos ett IC för att skatta glesa modellstrukturer (se nedan). I kapitel 5-7 behandlas sedan multimodellering. Multimodellering är ett sätt att, istället för att välja en enskild modell, vikta resultaten (till exempel parameterskattningar, eller prediktioner av framtida signalvärden) baserade på flera olika modellstrukturer. Vi diskutera varför multimodeller ofta är att föredra framför enskilda modeller.

I början av del I behandlas så kallad ”klassisk” modellselektion, medan kapitel 6 och 7, använder Bayesiansk teori. Kapitel 6 är mer applikationsinriktat än övriga kapitel, och behandlar modellering av okända kanaler för skattning av överförda symboler vid trådlös kommunikation. I kapitel 7 behandlas optimal modellselektion och parameterskattning för linjära regressionsmodeller.

Del II: Gles modellering

Den andra delen av denna avhandling, kapitel 8-11, behandlar glesa modeller. En gles modell kan bestå av en godtycklig kombination av en mängd parametrar. Detta innebär att vi, om vi betraktar totalt $n$ parametrar i en mängd möjliga modellfunktioner för $g(t)$, kommer att få $2^n$ möjliga modellfunktioner.\footnote{Detta eftersom varje möjlig modellstruktur kan beskrivas av ett binärt tal med $n$ siffror, där varje siffra motsvarar en parameter. Värdet 1 motsvarar då en ”aktiv” parameter, och värdet 0 motsvarar en parameter som är exkluderad från den aktuella modellen.} Glesa modeller spelar en viktig roll inom, till exempel, statistisk dataanalys, där det ofta är viktigt att veta vilka faktorer, av många möjliga, som mest sannolikt förklarar det uppmätt data. Andra viktiga applikationsfält finns inom kommunikation och radar, där ett fält starka reflektioner av den utsända signalen ofta dominerar i den mottagna signalen. Se, till exempel, kapitel 10 och 11.


I kapitel 8 presenterar vi en enkel algoritm för att skatta och kombinera glesa kanalmodeller. Denna metod kombinerar resultaten från kapitel 4 och 5
med antagandet om en vit träningssignal (det senare antagandet används för att minska beräkningskomplexiteten). Kapitel 9 behandlar optimal parameter-skattning och modellselektion under antagandet att signalen genererades av en gles, linjär modell.

### Glossary and Notation

#### Glossary and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>a posteriori</em></td>
<td>after having considered the data</td>
</tr>
<tr>
<td><em>a priori</em></td>
<td>before having considered the data (i.e., based on prior belief and experience)</td>
</tr>
<tr>
<td>AIC</td>
<td>Akaike Information Criterion (or Information Criterion A)</td>
</tr>
<tr>
<td>AIC$_c$</td>
<td>corrected AIC</td>
</tr>
<tr>
<td>AR</td>
<td>AutoRegressive</td>
</tr>
<tr>
<td>ARMA</td>
<td>AutoRegressive Moving Average</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BIC</td>
<td>Bayesian Information Criterion (or Information Criterion B)</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase-Shift Keying</td>
</tr>
<tr>
<td>CRB</td>
<td>Cramér-Rao Bound</td>
</tr>
<tr>
<td>dB</td>
<td>decibel: the transformation equation $10 \log_{10}(\cdot)$ is used for power values, whereas $20 \log_{10}(\cdot)$ is used for amplitude values. Unless anything else is specified, $0$ dB corresponds to unity</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>GCV</td>
<td>Generalized Cross-Validation</td>
</tr>
<tr>
<td>GIC</td>
<td>Generalized Information Criterion</td>
</tr>
<tr>
<td>GLRT</td>
<td>Generalized Likelihood Ratio Test</td>
</tr>
<tr>
<td>IC</td>
<td>Information Criterion</td>
</tr>
<tr>
<td>iff</td>
<td>if and only if</td>
</tr>
<tr>
<td>i.i.d.</td>
<td>independent and identically distributed</td>
</tr>
<tr>
<td>KL</td>
<td>Kullback-Leibler</td>
</tr>
<tr>
<td>LT</td>
<td>Long Term</td>
</tr>
<tr>
<td>LTV</td>
<td>Long Term Validation</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori</td>
</tr>
<tr>
<td>MDL</td>
<td>Minimum Description Length</td>
</tr>
<tr>
<td>MF</td>
<td>Matched Filter</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
</tbody>
</table>

17
MLSD  Maximum Likelihood Sequence Detector  
MMSE  Minimum Mean Square Error  
MRC  Maximum Ratio Combining  
MSE  Mean Square Error  
pdf  probability density function  
PSD  Power Spectral Density  
QPSK  Quadrature Phase-Shift Keying  
RMSE  Root Mean Square Error  
SNR  Signal-to-Noise Ratio  
ST  Short Term  
s.t.  subject to  
STV  Short Term Validation  
SV  Saleh-Valenzuela (an indoor radio channel model)  
VA  Viterbi Algorithm

### Notation

- **boldface lower case letters** are used for vectors
- **boldface upper case (capital) letters** are used for matrices
- **non-bold letters** are generally used to denote scalars
- \( \cdot^T \) stands for matrix or vector transpose
- \( \cdot^H \) stands for Hermitian (conjugate) transpose of a matrix or a vector
- A hat, \( \hat{\cdot} \), is used to denote an estimate
- \( \mathbb{C}^{n \times m} \) the complex \( n \times m \)-dimensional space
- \( \mathbb{C}^n \) the complex \( n \)-dimensional plane (\( \mathbb{C} \) is used for \( n = 1 \))
- \( \mathcal{CN}(\mu, \mathbf{R}) \) the complex-valued Gaussian (normal) distribution with mean \( \mu \) and covariance matrix \( \mathbf{R} \)
- \( \text{diag}\{\cdot\} \) diagonal; \( \text{diag}\{a\} \) means the matrix with the vector \( a \) on the diagonal and zeros everywhere else, and \( \text{diag}\{A\} \) means the vector containing the elements on the diagonal of the matrix \( A \)
- \( \mathbf{E}\{\cdot\} \) the expectation operator
- \( \exp(\cdot) \) the exponential function; \( \exp(a) = e^a \)
- \( i \) the imaginary unit, \( \sqrt{-1} \), unless otherwise specified
- \( \mathbf{I} \) the identity matrix (of unspecified dimension)
- \( \mathbf{I}_n \) the \( n \times n \) identity matrix
- \( \text{Im}(\cdot) \) the imaginary part of
- \( \ln(\cdot) \) the natural logarithm
- \( \mathcal{N}(\mu, \mathbf{R}) \) the real-valued Gaussian (normal) distribution with mean \( \mu \) and covariance matrix \( \mathbf{R} \)
- \( \mathcal{O}(\cdot) \) on the order of; used, e.g., to denote computational complexity: a computational complexity of \( \mathcal{O}(n) \) means that the required number of operations is proportional to \( n \)
GLOSSARY AND NOTATION

$p(a|b)$ the probability density function of a continuous variable $a$ conditioned on $b$

$P(a|b)$ the probability mass function of the discrete variable $a$ conditioned on $b$

$\mathbb{R}^{n \times m}$ the real $n \times m$-dimensional space

$\mathbb{R}^n$ the real $n$-dimensional plane ($\mathbb{R}$ is used for $n = 1$)

$\text{Re}(\cdot)$ the real part of

$\text{tr}(\cdot)$ the trace of a matrix

$\mathcal{U}(a, b)$ the uniform distribution between $a$ and $b$

$\sim$ distributed as; e.g., $x \sim \mathcal{N}(\mu, R)$ means that $x$ is Gaussian distributed with mean $\mu$ and covariance matrix $R$

$\in$ in; e.g., $x \in \mathbb{C}^n$ means that $x$ lies in the $n$-dimensional complex plane, and $X \in \mathbb{R}^{n \times m}$ means that $X$ is a real valued matrix with $n$ rows and $m$ columns

\setminus minus operation for sets; e.g., $\{1, 2, 3, 4, 5\} \setminus \{2, 4\} = \{1, 3, 5\}$

$:= $ set to; e.g., $a := b$ means that $a$ is set to $b$

$\triangleq$ defined as

$\cong$ equality up to irrelevant constants

$\mid$ conditioned on; e.g., $a|b$ means $a$ conditioned on $b$

$| \cdot |$ matrix determinant

$\| \cdot \|_k$ the $L_k$-norm; $\|a\|_k = \left( \sum_j |a_j|^k \right)^{1/k}$, $k \geq 1$, $a = [a_1, a_2, \cdots]^T$

$\| \cdot \|_0$ the $L_0$-norm; $\|a\|_0 = \text{“the number of non-zero elements of } a$$

$\| \cdot \|$ the $L_2$-norm (Euclidean norm)
Introduction

This thesis on signal processing concerns model based inference when the appropriate signal model structure is unknown. In this introductory chapter we will cover some background material, describe the contributions of the thesis, and discuss some topics for future research.

1.1 Why is the Topic of this Thesis Important?

The advances in signal processing (and related fields) have had a tremendous impact on everyday life of modern man. We rely on signal processing for communication over cellular phones and the Internet, for comfortably carrying our music collection in the mp3 or related formats, for interpreting medical data for diagnostics, for localization via the Global Positioning System (GPS), for terrain mapping and collection of meteorological data using radar, etc. (for just a few more of a multitude of possible signal processing applications, see Table 1.1). Many signal processing methods are based on parametric models of the signals (see below), and the performance of such methods depends heavily on the chosen model structure and on the quality of the parameter estimates. New results on parametric modeling can thus offer increased performance in a multitude of applications. Just a few possible improvements are faster data throughput and lower energy consumption for hand-held communication terminals, more accurate medical diagnoses, better weather forecasts, earlier and more reliable earthquake and tsunami warnings, etc.

1.2 Signal Processing

Signal processing deals with the representation, interpretation, manipulation and transformation of signals. The term signal is very general: in a broad definition a signal is an n-dimensional function over an m-dimensional domain (such as time, space, frequency, etc.). The perhaps most common type of
Table 1.1: Some examples of signals and applications of signal processing.

<table>
<thead>
<tr>
<th>Example of signal</th>
<th>Possible applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>temperature outside a building</td>
<td>learn the temperature periodicities, or predict what the temperature will be in one hour</td>
</tr>
<tr>
<td>the reading of a seismometer</td>
<td>predict earthquakes</td>
</tr>
<tr>
<td>the recorded tone of a guitar</td>
<td>determine the pitch (for tuning)</td>
</tr>
<tr>
<td>string</td>
<td></td>
</tr>
<tr>
<td>electrocardiogram of a patient</td>
<td>determine the pulse, or indicate whether the patient has a heart problem</td>
</tr>
<tr>
<td>voltage induced in the antenna of</td>
<td>transform the received signal into speech</td>
</tr>
<tr>
<td>a cellular phone</td>
<td></td>
</tr>
</tbody>
</table>

signal is the 1-dimensional signal over time. In this thesis, we only cover 1-dimensional signals. In Table 1.1, some examples of signals together with possible applications of signal processing are listed.

In the thesis we exclusively consider digital signal processing. This means that the originally continuous signals have been sampled (i.e., values of the signal have been recorded) at a number of points. The signals can thus be represented as sequences of numbers, where each number corresponds to a position in the domain (e.g., time) from which the signal was sampled.

A broad introduction to digital signal processing lies beyond the scope of the thesis. Instead, we refer the reader to the numerous textbooks on the topic, e.g., [61; 75; 76; 147].

1.3 Parametric Models in Signal Processing

The word model in the title of the thesis refers to a so-called parametric model of a signal. Parametric models offer an efficient way of signal representation and interpretation. A parametric model is a mathematical function that depends on the values of some parameters. The aim in parametric modeling is often to adjust the parameters of an appropriate model function such that the model optimizes some criterion, such as fitting the measured signal with a minimum possible error. This task is called parameter estimation. If the parametric model is derived from known properties of the physical process which generated the signal, then the accurate estimation of the parameters is often the final goal of the modeling task. Otherwise, the estimation of the parameters can be a means of achieving some other goal, such as finding the spectrum of a signal [147] or predicting future values [142]. Parametric models are also very effective

1However, some of the material left out of the thesis, namely that which deals with the beamforming application, considers 2-dimensional signals. See Section 1.7.
for data compression, since a parametric model can often describe the relevant properties of a signal in a very compact form.

The opposite of a parametric model is a non-parametric model. There is no common form for non-parametric models; they can be represented by graphs, step responses, Bode plots, spectra, etc. [140]. We will not treat non-parametric modeling in this thesis.

Let \( y(t) \) describe a signal sampled at the time instants \( t = 0, 1, \ldots, N - 1 \). Then \( y(t) \) can be modeled as follows:

\[
y(t) = f(\theta, t) + e(t),
\]

where \( f(\cdot) \) will be termed the parametric model function\(^3\), \( \theta \) is the parameter vector (some commonly used model functions and associated parameter vectors are listed in Table 1.2) and \( e(t) \) describes the difference between the measured signal and the model function.\(^4\) The term \( e(t) \) is often termed noise, error or residual. In practical applications of signal processing there will always be a non-zero noise term included in the signal, since there is no such thing as a perfect measurement. The specification and treatment of the (additive) noise \( e(t) \) is very important.

The usefulness of (1.1) depends critically on how well suited the expression \( f(\theta, t) \) and the assumptions on \( e(t) \) are for describing the data \( y(t) \). No matter how well-tuned the parameter vector \( \theta \) is, the results will be of little value, or even misleading, unless \( f(\theta, t) \) is appropriately chosen. Model selection is the art of selecting the most appropriate expression for \( f(\theta, t) \) (and distribution of \( e(t) \)) from several candidates. This will be detailed further below.

It is important to realize that \( f(\theta, t) \) is only a model of reality. The concept of a true model (i.e., a model exactly equal to the data-generating reality) is sometimes discussed. However, except in computer simulations or pathological cases, the reality which generate \( y(t) \) will always be far more complex than what is possible, or even desired, to describe in a compact function form \( f(\theta, t) \). In this light, it may be seen as problematic that several methods for parametric modeling are based on the (erroneous) assumption that the model structure of the true data-generating process is among the set of considered model structures (see Section 1.3.2). While it is sometimes claimed that this is a serious deficiency of many methods [19], we expect that methods relying on the above assumption should perform well, if the data-generating reality can be well approximated by models in the considered set.

---

\(^2\)We will generally let the time index \( t \) denote the sample number, and not the actual time instant (in seconds) at which the signal was sampled.

\(^3\)We use the expression \( f(\theta, t) \) for both stochastic and deterministic models (see below). In a stochastic model, \( f(\theta, t) \) may depend on earlier samples of the signal \( y(t) \) or the noise \( e(t) \).

\(^4\)Note that \( e(t) \) also normally depends on some parameters describing its properties (such as mean and variance). However, to simplify the notation, we omit writing out the parameter vector for \( e(t) \) herein.
Table 1.2: Some common signal models. In the table, $e(t)$ (also appearing in (1.1)) is a white, zero-mean noise sequence [147] with variance $\sigma^2$, $x(t)$ denotes a known input signal and $[X]_{t,k}$ denotes the element on the $t$th row and the $k$th column of a known regressor matrix $X$.

<table>
<thead>
<tr>
<th>Model name</th>
<th>Model function $f(\theta, t)$</th>
<th>Parameters $\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum of sinusoids</td>
<td>$\sum_{k=1}^{K} a_k \sin(\omega_k t + \phi_k)$</td>
<td>${a_k, \omega_k, \phi_k}_{k=1}^{K}$</td>
</tr>
<tr>
<td>sum of complex sinusoids</td>
<td>$\sum_{k=1}^{K} a_k e^{i(\omega_k t + \phi_k)}$</td>
<td>${a_k, \omega_k, \phi_k}_{k=1}^{K}$</td>
</tr>
<tr>
<td>sum of exponentially damped complex sinusoids</td>
<td>$\sum_{k=1}^{K} a_k e^{i(\omega_k t + \phi_k) - \alpha_k t}$</td>
<td>${a_k, \omega_k, \phi_k, \alpha_k}_{k=1}^{K}$</td>
</tr>
<tr>
<td>finite impulse response (FIR) filter</td>
<td>$\sum_{k=1}^{K} h_k x(t - k)$</td>
<td>${h_k}_{k=1}^{K}$</td>
</tr>
<tr>
<td>linear regression</td>
<td>$\sum_{k=1}^{K} [X]_{t,k} h_k$</td>
<td>${h_k}_{k=1}^{K}$</td>
</tr>
<tr>
<td>polynomial</td>
<td>$\sum_{k=1}^{K} \theta_k t^{k-1}$</td>
<td>${\theta_k}_{k=1}^{K}$</td>
</tr>
<tr>
<td>autoregressive (AR) process</td>
<td>$\sum_{k=1}^{K} a_k y(t - k)$</td>
<td>${a_k}_{k=1}^{K}, \sigma^2$</td>
</tr>
<tr>
<td>moving average (MA) process</td>
<td>$\sum_{k=1}^{K} b_k e(t - k)$</td>
<td>${b_k}_{k=1}^{K}, \sigma^2$</td>
</tr>
<tr>
<td>autoregressive moving average (ARMA) process</td>
<td>$\sum_{k=1}^{K} a_k y(t - k) + \sum_{j=1}^{J} b_j e(t - j)$</td>
<td>${a_k}<em>{k=1}^{K}, {b_j}</em>{j=1}^{J}, \sigma^2$</td>
</tr>
</tbody>
</table>

1.3.1 Model Characterization

There exist many different classes of parametric signal models. Some different types of models are listed and described below.

Physical models vs. black-box models

A physical model is derived from knowledge of the physical reality which generates the data. This means that the parameters are interpretable quantities, and their accurate estimation is often the task of interest. A simple example of a physical model is the expression for how Newton’s laws of mechanics [10] act on a body in constant acceleration (in the 1-dimensional case):

$$y(t) = y_0 + v_0 t + \frac{1}{2} a t^2 + e(t),$$

where the signal $y(t)$ is the (measured) position at time instant $t$, and $\theta = [y_0, v_0, a]^T$ is the parameter vector with $y_0$ being the position at $t = 0$, $v_0$ being the velocity at $t = 0$, and $a$ being the acceleration. The term $e(t)$ (which is commonly not seen in the physics books) describes measurement and modeling errors.

A black-box signal model, on the other hand, is not derived from any physical properties of the signal. It is simply a mathematical description that is appropriate for the signal under study, and the model parameters do not have
1.3. Parametric Models in Signal Processing

A physical meaning. The autoregressive (AR) process (see Table 1.2) is an example of a signal model which is often used as a black-box model. Note that the characterization of a model as a physical model or a black-box model is not a property of the model equation itself; rather it depends on the application in which the model is used. An AR model could be a physical model in some applications, where the data-generating reality is known to obey the AR equation, and a black-box model in other applications where little is known about the data-generating process.

Deterministic models vs. stochastic models

A deterministic model is a model for which, once the exact parameter values are known, the signal can be reproduced exactly. A stochastic model is inherently random, so exact signal reproduction is not possible. In Table 1.2, the first six model functions are examples of deterministic models, whereas the three last model functions are stochastic. Deterministic signal models are seldomly used alone in signal processing applications; rather, signals are often modeled as a deterministic part plus stochastic noise.

Linear models vs. non-linear models

A signal model is linear if the signal depends linearly on the model parameters and possible inputs and disturbances. Examples of linear models are the finite impulse response (FIR) filter, the linear regression, and the autoregressive (AR) model in Table 1.2. Examples of non-linear signal models are the sums of sinusoids in Table 1.2.

1.3.2 Model Selection

In some applications an appropriate model function \( f(\theta, t) \) is known beforehand. For example, it would be difficult to motivate the use of any function different from that in (1.2) for describing the position of a macroscopic body in constant acceleration in free space at non-relativistic velocities. However, often only an appropriate function type is known, and not the exact expression. In those instances, there will be several candidate model functions to choose from. The goal in model selection is to decide which one of the candidates is most appropriate for describing \( y(t) \).

Example: Magnetic Resonance Spectroscopy. In magnetic resonance spectroscopy, it is known from the underlying physics that a sum of exponentially damped complex sinusoids (see Table 1.2) in noise is an appropriate signal model [124]. However, the number of sinusoids in the data is often unknown. This means that we do not have a single model function \( f(\theta, t) \) for describing \( y(t) \). Instead, we have a number of, say \( M \), candidate model functions, \( \{f_m(\theta_m, t)\}_{m=1}^M \), where each model function describes a sum of a specific number of exponentially damped complex sinusoids.

\(^5\)Note that the noise \( e(t) \) enters the AR model function via \( y(t) \).
At first sight, it seems tempting to select a very complex model (a model with many parameters) for describing $y(t)$. Such a model is highly flexible, and it can likely describe most of the characteristics of the signal. Indeed, if the considered models have nested structures (see below), the models with more parameters will always fit the measured data better than the models with fewer parameters (due to their increased degrees of freedom). However, there are (at least) two problems with this naïve approach: First, the more complex model we choose, the more data we need to estimate its parameters accurately. If we have only 10 data samples we might be able to estimate the coefficients of a 0th or 1st degree polynomial with a reasonable accuracy, but hardly the coefficients of a polynomial of degree 8, unless the noise level is very low. Therefore, for a fixed number of data samples, we must not choose a model that is too complex. Otherwise, the parameter estimates will often be severely affected by the random measurement noise (so called overfitting). Second, for the purpose of interpretation we may wish to exclude, from the model function, parameters which are unlikely to have played any role in the generation of $y(t)$.

The first of the above two problems is an example of the seemingly ever present bias-variance compromise [147]: For a highly complex model the ratio between the number of available samples for estimation and the number of parameters to estimate will be relatively low, so the variance of the parameter estimates becomes large. On the other hand, a model of low complexity, which has a high sample length to number of parameters ratio, may not be able to model the full dynamics of the data-generating process. The parameter estimates will then often be biased.

**Example: Polynomial fitting.** In Figure 1.1 we show a simple illustration of the bias-variance compromise via an example of estimation of polynomial coefficients. Nine data samples (marked with +−-marks in the figure) were generated by a third degree polynomial in noise: $y(x) = 0.1 - 2x + 0.6x^2 + 2x^3 + e(x)$, where $e(x)$ is white, zero-mean Gaussian noise with variance 4. The coefficients of three polynomial models of different degrees were estimated (by the method of least squares [140]) using the 9 data samples. As can be seen from the figure, the polynomial model of degree 2 is not flexible enough to accurately model the data-generating process (the estimate is biased). The polynomial model of degree 8 follows the data points perfectly, but does not look like the true polynomial for other values of $x$ (the estimate has a high variance). This is an example of overfitting, where the extra degrees of freedom offered by parameters not present in the data-generating process are used to model the (non-interesting) noise in the data. The curve of the polynomial model of the correct degree does not pass exactly through the (noisy) data points, but on the average it offers the best description of the true underlying (noise-free) polynomial. A good model selection algorithm should prefer this third degree model to the other two.
Figure 1.1: Estimation of polynomial models of different degrees.
The Occam’s razor principle is often discussed in relation to the problem of model selection. This principle of parsimony, as it is sometimes denoted, states essentially that \textit{all things being equal, the simplest solution should be preferred}. For model selection purposes, this should be interpreted as follows: \textit{When several models are equal in other respects, the model which imposes the least restrictive assumptions and introduces the fewest parameters should be selected.} This is a reasoning which is intuitively pleasing. It is also interesting to note that this is essentially how sound model selection algorithms work: they often include a term which penalizes complex models (see Chapter 2; also Bayesian model selection can be interpreted as having an \textit{Occam factor} [92] which works in a similar manner).

1.3.3 Nested Models

A set of models is nested when each model in the set can be described as a special case of the models in the set with higher complexity. Examples of nested models are polynomials of different degrees (a polynomial of degree $m$ can be described as a polynomial of degree $n > m$ for which all coefficients corresponding to degrees $> m$ are equal to zero), autoregressive (AR) or moving average (MA) models of different orders, finite impulse response (FIR) filters of different lengths, etc. [147]. In these cases, a single integer valued parameter is sufficient to describe the model complexity. This integer valued parameter is often denoted by \textit{order} $^7$, and since the order completely specifies the model structure when the models are nested, model selection for nested models is often called \textit{model order selection}.

The first part of the thesis is primarily concerned with nested models. More specifically, the first part deals with sets of models which are relatively small; see below. In Chapter 2 some classical and general methods for model selection are reviewed. These methods are known as information criteria (IC), and their application consists of the computation of a criterion for each model in the set. The model which minimizes this criterion is then selected. While nothing prevents us from using such methods also for sparse model selection, we then quickly run into computational difficulties since the number of sparse model structures grows exponentially with the number of parameters (see below). The IC are general in the sense that they can be applied to any set of parametric models provided the maximum likelihood (ML) estimate of the parameters can be obtained. For many types of models the ML estimates are difficult to obtain and thus more specialized methods, tied to specific model types, have been suggested. Some examples include methods for ARMA order selection [43] and methods for estimation of the number of sinusoids in noise [30; 44].

$^6$The principle of Occam’s razor is attributed to the 14th century English logician and Franciscan friar William of Occam. The initial formulation was \textit{entia non sunt multiplicanda praeter necessitatem}, or \textit{entities should not be multiplied unnecessarily}.

$^7$\textit{Order} usually refers to the number of parameters in the model (see Chapter 2).
1.3.4 Sparse Models

The second part of the thesis is devoted to the study of sparse models, specifically sparse linear models. In the general sparse model set, a model can consist of any combination of the considered parameters. This means that if we consider $n$ parameters we get $2^n$ possible model structures.\footnote{This is because each model structure can be described by a binary number with $n$ digits (where each digit represents a parameter), where a 1 indicates an “active” parameter, and a 0 means that the parameter is excluded from the model.} Prior knowledge about the allowed model structures can, however, reduce this number considerably. Sparse models play an important role when it is desired to learn which few factors, out of several possible, are most likely to explain the data (see, e.g., Chapter 9). Non-linear models can also often be formulated as linear sparse models (see, e.g., [94]). In some active sensing applications, such as deconvolution of ultrasonic pulse-echo signals [104], non-destructive testing [103; 105], reflection seismology [79] and radar (see, Chapter 11), the data can often be well described by sparse linear models. The general problem of interest is then detection and estimation of time-delayed reflections of the probing signal [45]. In Chapters 8 and 10 we treat sparse linear models for symbol detection in a communications application.

It is clear that, unless the number of considered parameters, $n$, is very small, it is computationally prohibitive to, as discussed in the previous section, compute a criterion for each of the $2^n$ sparse model structures. There are a few different suggestions for how to deal with this problem available in the literature.

One option is to construct an estimator of the full model (i.e., the model containing all parameters one chooses to consider) which has the property of shrinking some parameter estimates to zero, thereby determining the model structure and the parameter values jointly. This is perhaps the most common approach in the literature. Some typical and well-known examples of this type of methods, designed for linear regression models, are Lasso (least absolute shrinkage and selection operator) [157] and Basis Pursuit [22].

Another approach is to reduce the number of models in the set from $2^n$ to some feasible number, and then base the inference on this reduced set of models. This reduction can be performed, e.g., by backward elimination (in which parameters are removed from consideration one-by-one, starting from the model containing all parameters) or by forward selection (in which parameters are added one-by-one, starting from a noise-only model) [165]. This type of approach is used, e.g., in Chapters 8, 9 and 11.

Yet another option is to iteratively add parameters to or remove parameters from the model until a certain stopping criterion is fulfilled. An example, for the total least squares model, is described in [48]. Another such approach, for linear Bernoulli-Gaussian models (the same type of mixture models that we use in Chapters 9 and 11), is described in [79]. A genetic algorithm of this type, for Bernoulli-Gaussian models, is explored in [106].
Aims of Sparse Modeling

In this thesis we focus on sparse modeling in applications for which the data-generating process itself can be assumed to be well described by a sparse model. The goal is then often to select the model structure which is most strongly supported by the data, or to find the parameter estimates which most accurately describe the data-generating process.

In other applications of sparse modeling, the goal is signal representation; i.e., to describe a measured signal vector, \( \mathbf{y} = [y(1) \cdots y(N)]^T \) using a linear combination of as few components as possible from a (known) dictionary \( \mathbf{X} \) of \( n > N \) signals (the components, or atoms, in the dictionary are in the columns of \( \mathbf{X} \), so \( \mathbf{X} \) is \( N \times n \)). Mathematically, this problem can be formulated as solving \( \mathbf{y} = \mathbf{Xh} \) where \( \mathbf{h} \) should have as few non-zero coefficients as possible (be the sparsest solution), i.e.,

\[
\hat{h}_0 = \arg \min_{\mathbf{h}} \| \mathbf{h} \|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{Xh}.
\] (1.3)

Solving the above \( L_0 \)-norm minimization problem in the general case is NP-hard, and the problem can only be solved by a combinatorial approach [32; 46]. Therefore it is common to instead consider the following \( L_1 \)-norm minimization, which can be viewed as an approximation of (1.3),

\[
\hat{h}_1 = \arg \min_{\mathbf{h}} \| \mathbf{h} \|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{Xh}.
\] (1.4)

The above problem can be transformed into a linear program which is straightforward to solve [46]. Interestingly, and surprisingly, the solutions of (1.3) and (1.4) are the same for a large class of underdetermined linear systems [32; 46]. These are two strong motivations for using the approximation (1.4) of (1.3).

The problem (1.3) is a fascinating one with a rich literature. However, we again remark that this is not the problem of primary interest when we discuss sparse modeling. The problem of our primary concern for estimation of \( \mathbf{h} \), which we explore in the Chapters 8, 9 and 11, can be loosely formulated as

\[
\hat{\mathbf{h}} = \mathbb{E}\{\mathbf{h}|\mathbf{y}\} \quad \text{conditioned on} \quad \mathbf{y} \text{ being generated by a sparse model plus noise}
\] (1.5)

where \( \mathbb{E}\{\cdot\} \) denotes the expectation operator (the details of the actual problems will be described in the relevant chapters). In this sense, we decouple the problems of sparse model selection and the problem of estimating \( \mathbf{h} \) under the assumption that it is sparse. Indeed, due to the way we model sparsity, the estimates of \( \mathbf{h} \) that we derive in Chapters 8, 9 and 11 will generally turn out to be non-sparse (even though many elements will often be close to zero), whereas the model selection algorithms developed in those chapters generally return sparse models.

Also, methods such as Lasso and its many relatives deal with problems similar in spirit to (1.5) rather than (1.3) (e.g., they assume that the data \( \mathbf{y} \) is noisy whereas (1.3) uses noise-free data), although more weight is often placed on obtaining sparse estimates rather than small expected errors of the parameter vector \( \mathbf{h} \) (which is the primary interest in this thesis). A general
problem description for those methods could be

\[
\hat{h} = \arg \min_{h} \| y - Xh \| \quad \text{subject to } h \text{ being sparse}
\]

where the sparsity can be enforced in different ways, such as adding a penalty which depends on \( h \) to the above equation, or by constraining a norm of \( h \). For some interesting results on when a solution corresponding to the sparsest solution in the noise-free case can be recovered from noisy data, we refer to [47].

1.3.5 The Use of a Model

Prior to making inferences based on parametric models, one should always consider the intended use of the models. Depending on whether the final goal is interpretation, prediction, estimation, detection, etc., different problems are to be solved. The often seen approach of first selecting a model, estimating its parameters, and finally basing the inferences on that model only is generally not optimal, although it may offer a good compromise between computational convenience and quality of the inference.

Model selection is essentially a detection problem in which the goal is to detect which model in a set is the most appropriate one for describing a given signal. If the model is interesting per se, which is often the case for physical models (see, Section 1.3.1) or in detection-based applications such as radar (under some assumptions, see Chapter 11), then model selection is the problem to consider.

In many applications, the actual model structure is not of interest. Examples include prediction of a variable (see Chapter 5), estimation of range profiles in radar (see Chapter 11), frequency function estimation [63], etc. There, model selection followed by inferences based on the selected model is not optimal. Rather, the contributions from the hypothesized models should be weighted together in a multi-model framework, as discussed in Chapter 5. Multi-model, or model averaging, approaches for parameter estimation are developed and discussed in Chapters 5, 7, 8, 9 and 11. Generally, one might expect that accurately estimated parameters result in well-performing models, even when parameter estimation is not the final goal of the modeling.

In Chapters 6 and 10 we consider symbol detection in communication applications. In such applications, neither the model structure nor the parameter values are the quantities of primary interest (the transferred symbols are), although accurate parameter estimation can often provide good (but non-optimal) symbol detection in a computationally efficient way (see Chapters 6 and 8).

1.4 Bayesian Inference

Bayesian inference is a scientific method for finding probabilities (or probability densities) of propositions by combining measured data and information given by the user. Bayesian arguments will be frequently used, especially in the later
parts of this thesis, so it seems worthwhile to provide a short background. For a thorough discussion on the theory and motivation behind Bayesian inference we refer to [69; 92].

The mechanisms of Bayesian inference operate on distributions and follow a few simple rules from probability theory. The big advantage of the Bayesian framework is that, once the necessary distributions are available, inference is a purely mechanical process which leads to optimal solutions (conditional on the information supplied by the user). The disadvantage is that these solutions are often computationally prohibitive, so approximations and/or numerical methods have to be used.

A great source of debate regarding the Bayesian framework concerns its dependency on a priori distributions, or priors, for the propositions. These priors reflect the user’s knowledge about a proposition before any data has been considered. Many researchers advocating the so-called classical methods of inference oppose the use of priors. They prefer to regard, e.g., a parameter as an unknown but deterministic value to which it is meaningless to assign a distribution as if it were random.\footnote{Note, however, that the Bayesian need not consider a variable as random just because he assigns a distribution to it. Instead, this distribution is used to express the Bayesian’s state of knowledge about that variable prior to observing the data.} Furthermore, it is often argued that assigning a prior to a proposition means that the resulting inference will be affected by the subjective prior decided by the user.

We will not pursue this debate here. We simply note that both the classical and the Bayesian frameworks of inference have their respective merits, and the users should try to be pragmatic and choose the framework most appropriate for their application.

To perform Bayesian inference, essentially only two tools are required: namely the product rule and marginalization. The first of these two tools has the following expression:

\[
p(A, B | C) = p(A | C)p(B | A, C). \tag{1.6}
\]

The above equation should be read “given the information \(C\), the probability for the propositions \(A\) and \(B\) to both be true equals the product between the probability that \(A\) is true and the probability that \(B\) is true given that \(A\) is true.” Since the propositions \(A\) and \(B\) are exchangeable, the product rule can also be written as

\[
p(A, B | C) = p(B | C)p(A | B, C) \tag{1.7}
\]

and by combining (1.6) and (1.7) and rearranging the terms, one obtains the familiar Bayes’ theorem:

\[
p(A | B, C) = \frac{p(A | C)p(B | A, C)}{p(B | C)}. \tag{1.8}
\]

Bayes’ theorem is often used to make a hypothesis and data “exchange positions” in \(p(\cdot)\) (since it shows how \(p(A | B, C)\) relates to \(p(B | A, C)\)). Say that \(A\)
stands for a hypothesis and \( B \) for some data. Then (1.8) tells us the relation between the probability of the hypothesis \( A \) given the data \( B \) and the probability that the data \( B \) is observed assuming the hypothesis \( A \) is true. The expression \( p(A|C) \) is the prior of the hypothesis \( A \) and \( p(B|C) \) is the prior of the data \( B \).

The second tool, i.e., marginalization, describes a way to get rid of nuisance parameters which are of no specific interest for the inference. For a proposition \( B \) that takes on discrete values:

\[
p(A|C) = \sum_B p(A, B|C).
\] (1.9)

In the above expression, \( B \) is the set containing every possible value of \( B \), so (1.9) should be read “the probability for \( A \) to be true equals the sum, over all possible values of \( B \), of the probabilities that both \( A \) and \( B \) are true.” If \( B \) is continuous, then the expression corresponding to (1.9) becomes

\[
p(A|C) = \int_B p(A, B|C)dB.
\] (1.10)

The Bayesian essentially get his desired information by repeated use of the tools (1.6) (and (1.8) derived from (1.6)) and (1.9) or (1.10).

1.5 The Gaussian Distribution

If a vector \( z \) of \( n \) real-valued elements has a Gaussian distribution (or a normal distribution, as it is often called) with mean \( \mu \) and covariance matrix \( R \), this will be stated as

\[
z \sim \mathcal{N}(\mu, R)
\] (1.11)

and the pdf (probability density function) of \( z \) takes the form

\[
p(z|\mu, R) = \frac{1}{(2\pi)^{n/2}|R|^{1/2}} \exp \left( -\frac{1}{2} (z - \mu)^T R^{-1} (z - \mu) \right).
\] (1.12)

If \( z \) were instead complex valued, but otherwise as above, the assumption that \( z \) is complex Gaussian will be denoted by

\[
z \sim \mathcal{CN}(\mu, R)
\] (1.13)

and the pdf of \( z \) then takes the form

\[
p(z|\mu, R) = \frac{1}{\pi^n |R|} \exp \left( -(z - \mu)^H R^{-1} (z - \mu) \right).
\] (1.14)

The use of the Gaussian distribution is omnipresent in the signal processing literature, and also in numerous other fields of applied mathematics. It is also frequently assumed in this thesis that unknown quantities, such as noise and linear regression coefficients, are Gaussian distributed. What are then the motivations for this assumption? As it turns out, there are many good reasons for assuming Gaussianity. We list below some of the most convincing motivations (see also Chapter 7 of [69] for a much more detailed discussion along these lines, and for several motivating derivations of the Gaussian distribution).
1.5.1 The Central Limit Theorem

The Central Limit Theorem is amongst the most important and celebrated results in statistics and probability theory. It states that the distribution of a sum of \( n \) independent and identically distributed random variables having a limited variance approaches the Gaussian distribution as \( n \to \infty \) \[69\]. Since much of the noise seen in signal processing applications originates from a combination of many small contributions (e.g., when individual electrons collide with atoms and generate tiny pulses of electromagnetic waves, which sum up to the Gaussian noise observed in many electrical components \[69\]) the Gaussian assumption on the noise can often be motivated by the Central Limit Theorem.

1.5.2 The Maximum Entropy Principle

In information theory, the entropy measures the uncertainty of a random variable. The Maximum Entropy Principle \[69\] essentially states that, when assuming a distribution for an unknown variable, one should always select the distribution which gives the maximum uncertainty (i.e., the maximum entropy distribution) subject to the constraints of the available information. Use of this principle guarantees that no assumptions are made which are not supported by the available information.

When the only things known about a random variable are its mean and its variance, then the maximum entropy distribution for that random variable turns out to be the Gaussian distribution with the given mean and variance \[69\]. In other words, to assume a Gaussian distribution for such a variable means that the information on the mean and variance is fully taken into account, but that care is taken not to impose any other assumptions on the variable. This motivation is different in spirit from that of the Central Limit Theorem: The Central Limit Theorem discusses if a random variable is likely to be Gaussian distributed, whereas the Maximum Entropy Principle discusses how to best express the state of knowledge regarding the value of a variable.

1.5.3 Computational Convenience

The mathematical properties of Gaussian distributions are attractive. For example, Gaussian distributions are preserved under linear transformations: If \( z \sim N(\mu, R) \), then \( v = Az + b \) is also Gaussian distributed, \( v \sim N(A\mu + b, ARA^T) \) \[140\]. Furthermore, optimal estimates (such as the conditional mean) generally turn out to be simple to compute when all distributions involved are Gaussian \[142\].

1.6 Thesis Outline

The main content of this thesis is divided into two parts. The first part, comprising Chapters 2-7, is named *Non-Sparse Modeling*. This part deals with
parametric modeling using relatively small sets of models (such as nested models, see Section 1.3.3). The second part is named, and deals with, Sparse Modeling. There is also a third part containing appendices.

Here follows an outline of the content and the publications included in this thesis. The main contributions of each chapter are briefly presented, together with references to the papers upon which the chapters are based.

The thesis follows the custom of previous theses from the Division of Systems and Control at Uppsala University. This means that the chapters correspond very closely to the papers which they are based upon, but that the thesis is organized as a monograph. Therefore, since all chapters have been written to be self-contained, there may be some overlap between the different chapters. Also, the notation may sometimes vary between the different chapters, but is always consistent within a chapter.

**Part I: Non-Sparse Modeling**

**Chapter 2: A Review of the AIC, GIC and BIC Rules**

The chapter presents, in a tutorial manner, the Information Criterion (IC) approach to model order selection. Some well known IC are derived and discussed. Chapter 2 is based on [150]:


**Chapter 3: Cross-Validation Rules for Model Order Selection**

In this chapter, two previously introduced cross-validation based rules for model order selection are presented and compared to the classical AIC and BIC/MDL rules. Chapter 3 is based on [148]:


**Chapter 4: On Information Criteria and the Generalized Likelihood Ratio Test of Model Selection**

Here, an equivalence between the Generalized Likelihood Ratio Test (GLRT) and the IC rules is demonstrated and discussed. As a corollary, a way of exploiting the local behavior of an IC for selecting the structure of sparse models is introduced and evaluated numerically. Chapter 4 is based on [152]:

Chapter 5: Multi-Model Approach to Model Selection

This chapter presents the multi-model approach to model selection, in which several candidate models are estimated and combined in a weighted manner, the weights being estimates of the \textit{a posteriori} probabilities of the models. Via some numerical examples it is shown that this multi-model approach offers better predictive performance than the corresponding single model approaches. Chapter 5 is based on [151]:


Chapter 6: Adaptive Equalization for Frequency-Selective Channels of Unknown Length

In this chapter the multi-model (or model averaging) approach is applied to a practical example; that of adaptive equalization for time-dispersive communication channels for which the impulse responses have unknown length. An optimal receiver is derived, under some assumptions, along with some computationally efficient approximations of it. Via numerical simulations these new receivers are shown to outperform conventional adaptive Viterbi equalizers that use a fixed or estimated channel length. Chapter 6 is based on [86; 87]:


Chapter 7: Empirical Bayes Linear Regression with Unknown Model Order

This chapter studies maximum \textit{a posteriori} probability model order selection and minimum mean square error coefficient vector estimation for linear regression models, assuming Gaussian distributed noise and coefficient vectors. This model selection approach and coefficient estimation approach both depend on user parameters describing the distributions which presumably generated the data. We also derive empirical Bayesian versions of the methods, which estimate the user parameters from the data and thus require little or no information from the user. Chapter 7 is based on [129; 131; 132]:


- Y. Selén and E. G. Larsson, “Empirical Bayes linear regression with unknown model order,” in *International Conference on Acoustics, Speech,


Part II: Sparse Modeling

Chapter 8: A Model Averaging Approach for Sparse Communication Channel Equalization

In this short chapter, a method is presented which estimates several sparse channels and averages them in a multi-model manner, thereby combining the ideas of Chapters 4 and 5. Under the assumption of a white training signal this can be done in a computationally efficient way. The method is evaluated in a communication context, similar to that in Chapter 6, and is shown to perform well when the channels are sparse. Chapter 8 is based on [134]:


Chapter 9: Linear Regression With a Sparse Parameter Vector

Here we consider the linear regression problem under the assumption that the coefficient vector is sparse such that it corresponds to a Bernoulli-Gaussian sequence [79]. Using a Bayesian framework we derive the minimum mean-square error estimator of the coefficient vector and a computationally efficient approximation of it. We also derive an empirical Bayesian version of the estimator, which does not require the selection of any user parameters. As a byproduct, we obtain a powerful model (“basis”) selection tool for sparse models. Chapter 9 is based on [83; 84]:


Chapter 10: RAKE Receiver for Channels with a Sparse Impulse Response

In this chapter we derive the optimal RAKE receiver for channels with a sparse impulse response. The receiver is Bayesian and requires the knowledge of certain user parameters. However, we also derive an empirical Bayesian version
of our receiver, which estimates the user parameters from the data. Chapter 10 is based on [128; 133]:


Chapter 11: Estimation of Semi-Sparse Radar Range Profiles

This final chapter extends the method derived in Chapter 9 from strictly sparse models to semi-sparse models. A semi-sparse model has a few coefficients which are significantly larger than the rest (for a strictly sparse model, the coefficients are either zero or non-zero). This type of models is often appropriate for estimation of range profiles in radar and other active sensing applications. In such applications, each coefficient in the range profile corresponds to a certain range bin in the illuminated area. If a range bin contains a target, the reflections from that bin will result in a value of the corresponding coefficient which is significantly larger than the value corresponding to a target-free range bin. We also derive a MAP (maximum a posteriori) target detector that does not require the choice of any detection threshold. Chapter 11 is based on [135]:


Part III: Appendices

Appendix A: Cyclic Minimizers, Majorization Techniques, and the Expectation-Maximization Algorithm

This appendix covers some iterative approaches for minimizing possibly non-linear functions. For example, the information criterion approach of model selection, described in Chapter 2, requires the maximum likelihood estimates of the parameter vectors of the considered models. The approaches described here can be used to obtain such estimates. Appendix A is based on [149]:


Appendix B: Proof of (3.22) (Chapter 3)

This appendix contains a proof of the claim (3.22) in Chapter 3.
Appendix C: Minimum Error Probability Threshold (Chapter 11)

In this appendix a threshold value for minimum error probability target detection (under some simplifying assumptions) that is used in Chapter 11 is derived.

1.7 Beside the Thesis

Apart from the work presented in this thesis, I have been involved in other research projects. These projects include magnetic resonance spectroscopy [124] and beamforming [2]. For conciseness reasons and for the purpose of obtaining a unified thesis, this material had to be excluded from the thesis. Nevertheless, I list the excluded material below, for the interested reader:


\(^{11}\)Student paper award winner: Honorary mention
1.8 Beyond the thesis

Science is always an ongoing process, which builds upon old results, explanations and experiments to find new solutions and interpretations. The present thesis should be seen in this light, and there is thus room for improvement of the results, or adaptation to new settings. Personally, I regard the contributions on Bayesian parameter estimation, model selection, and symbol detection to be the most interesting parts of my work, and I would like to explore more the potential of such approaches for other types of signal models.

One thing not considered here is the evaluation of the suggested approaches on real (as in non-simulated) data. The problem with such data is often that there is no known truth, and it is therefore generally difficult to use real data for performance assessment. However, for the symbol detection applications discussed in Chapters 6 and 10, it would be possible to build a testbed and measure bit error rates. Also the target detection algorithm described in Chapter 11 should be possible to evaluate on real data.

In many applications of sparse modeling, such as the symbol detection discussed in Chapter 10, the semi-sparse setting described in Chapter 11 is likely a “better” description of reality than the strictly sparse setting (provided good values for the \( a_{\text{priori}} \) parameters can be found). It would be interesting to generalize the receiver in Chapter 10 to the semi-sparse setting.

While the radar model used in Chapter 11 is appropriate for static scenarios or relatively slowly moving targets it may be poor if the targets move fast compared to the wavelengths of the probing signal. For such cases, the Doppler effect shows up as errors in the regressor matrix containing the probing signal (i.e., \( S \) (11.1)). While the treatment of such errors is conceptually straightforward in the suggested Bayesian MMSE framework, the resulting expressions are likely too complex for direct use. An interesting extension of the work in Chapter 11 would be to treat the Doppler effect in a sound, yet computationally efficient, manner.
Part I

Non-Sparse Modeling
Chapter 2

A Review of the AIC, GIC and BIC Rules

2.1 Introduction

The parametric (or model-based) methods of signal processing require often not only the estimation of a vector of real-valued parameters but also the selection of one or several integer-valued parameters that are equally important for the specification of a data model. Examples of these integer-valued parameters of the model include the orders of an autoregressive moving average model, the number of sinusoidal components in a sinusoids-in-noise signal, and the number of source signals impinging on a sensor array. In each of these cases, the integer-valued parameters determine the dimension of the parameter vector of the data model, and they must be estimated from the data.

In what follows we will use the following symbols:

\[
\begin{align*}
y & = \text{the vector of available data (of size } N) \\
\theta & = \text{the (real-valued) parameter vector} \\
n & = \text{the dimension of } \theta.
\end{align*}
\]

For short, we will refer to \( n \) as the model order, even though sometimes \( n \) is not really an order (see, e.g., the above examples). We assume that both \( y \) and \( \theta \) are real-valued:

\[
\begin{align*}
y & \in \mathbb{R}^N \\
\theta & \in \mathbb{R}^n.
\end{align*}
\]
Whenever we need to emphasize that the number of elements in \( \theta \) is \( n \), we will use the notation \( \theta^n \). A method that estimates \( n \) from the data vector \( y \) will be called an order selection rule. Note that the need for estimating a model order is typical of the parametric approaches to signal processing. The non-parametric methods do not have such a requirement.

The literature on order selection is as considerable as that on (real-valued) parameter estimation (see, e.g., [19; 23; 90; 96; 122; 140; 145] and the many references therein). However, many order selection rules are tied to specific parameter estimation methods and hence their applicability is rather limited. Here we will concentrate on order selection rules that are associated with the maximum likelihood (ML) method of parameter estimation. The ML method is likely the most commonly used parameter estimation method. Consequently, the order estimation rules that can be used with the ML method are of quite a general interest.

2.2 Maximum Likelihood Parameter Estimation

In this section we review briefly the ML method of parameter estimation and some of its main properties that are of interest in this thesis. Let

\[
p(y|\theta) = \text{the probability density function (pdf) of the data vector } y,
\]

which depends on the parameter vector \( \theta \); also called the likelihood function.

The ML estimate of \( \theta \), which we denote by \( \hat{\theta} \), is given by the maximizer of \( p(y|\theta) \), see, e.g., [8; 15; 59; 108; 109; 112; 126; 140; 156]. Alternatively, as \( \ln(\cdot) \) is a monotonically increasing function,

\[
\hat{\theta} = \arg \max_{\theta} \ln p(y|\theta).
\] (2.1)

Under the Gaussian data assumption, ML typically reduces to the nonlinear least-squares (NLS) method of parameter estimation. To illustrate this fact, let us assume that the observation vector \( y \) can be written as

\[
y = \mu(\gamma) + e \tag{2.2}
\]

where \( e \) is a (real-valued) Gaussian white noise vector with mean zero and covariance matrix given by \( E\{ee^T\} = \sigma^2 I \). \( \gamma \) is an unknown parameter vector, and \( \mu(\gamma) \) is a deterministic function of \( \gamma \). It follows readily from (2.2) that

\[
p(y|\theta) = \frac{1}{(2\pi)^{N/2}(\sigma^2)^{N/2}} e^{-\frac{\|y-\mu(\gamma)\|^2}{2\sigma^2}} \tag{2.3}
\]

where

\[
\theta = \begin{bmatrix} \gamma \\ \sigma^2 \end{bmatrix} \tag{2.4}
\]

We deduce from (2.3) that

\[
-2\ln p(y|\theta) = N \ln 2\pi + N \ln \sigma^2 + \frac{\|y-\mu(\gamma)\|^2}{\sigma^2} \tag{2.5}
\]
A simple calculation based on (2.5) shows that the ML estimates of \( \gamma \) and \( \sigma^2 \) are given by

\[
\hat{\gamma} = \arg \min_{\gamma} \| y - \mu(\gamma) \|^2
\]

\[(2.6)\]

\[
\hat{\sigma}^2 = \frac{1}{N} \| y - \mu(\hat{\gamma}) \|^2.
\]

\[(2.7)\]

The corresponding value of the likelihood function is given by

\[-2 \ln p(y|\hat{\theta}) = \text{constant} + N \ln \hat{\sigma}^2.\]

\[(2.8)\]

As can be seen from (2.6), in the present case ML indeed reduces to NLS.

A special case of (2.2), which we will deal with in this chapter, is the sinusoidal signal model:

\[
y_s(t) = \sum_{k=1}^{n_s} \alpha_k e^{i(\omega_k t + \phi_k)} + e(t), \quad t = 1, \ldots, N_s
\]

\[(2.9)\]

where \([\alpha_k, \omega_k, \phi_k]\) denote the amplitude, frequency and phase of the \(k\)-th sinusoidal component, \(N_s\) is the number of observed complex-valued samples, \(n_s\) is the number of sinusoidal components present in the signal and \(e(t)\) is the observation noise. In this case

\[
N = 2N_s
\]

\[
n = 3n_s + 1.
\]

\[(2.10)\]

\[(2.11)\]

We will use the sinusoidal signal model in (2.9) as a vehicle for illustrating how the various general order selection rules presented in what follows should be used in a specific situation.

Next, we note that under regularity conditions the pdf of the ML estimate \(\hat{\theta}\) converges, as \(N \to \infty\), to a Gaussian pdf with mean \(\theta\) and covariance equal to the Cramér-Rao Bound (CRB) matrix (see, e.g., [91; 140] for a discussion about the CRB). Consequently, asymptotically in \(N\), the pdf of \(\hat{\theta}\) is given by

\[
p(\hat{\theta}) = \frac{1}{(2\pi)^{n/2}|J^{-1/2}|^{1/2}} e^{-\frac{1}{2}(\hat{\theta} - \theta)^T J(\hat{\theta} - \theta)}
\]

\[(2.12)\]

where

\[
J = -E \left\{ \frac{\partial^2 \ln p(y|\theta)}{\partial \theta \partial \theta^T} \right\}
\]

\[(2.13)\]

is the so-called (Fisher) information matrix.

**Remark:** To simplify the notation, we use the symbol \(\theta\) for both the true parameter vector and the parameter vector viewed as an unknown variable. The exact meaning of \(\theta\) should be clear from the context. 

The “regularity conditions” referred to above require that \(n\) is not a function of \(N\), and hence that the ratio between the number of unknown parameters
and the number of observations tends to zero as $N \to \infty$. This is true for most parametric signal processing problems, but not for all (see e.g. [107; 144; 161]).

To close this section, we note that under mild conditions:

$$\left[ -\frac{1}{N} \frac{\partial^2 \ln p(y|\theta)}{\partial \theta \partial \theta^T} - \frac{1}{N} J \right] \to 0 \text{ as } N \to \infty. \quad (2.14)$$

To motivate (2.14) for the fairly general data model in (2.2) we can argue as follows. Let us rewrite the negative log-likelihood function associated with (2.2) as (see (2.5))

$$-\ln p(y|\theta) = \text{constant} + \frac{N}{2} \ln \sigma^2 + \frac{1}{2\sigma^2} \sum_{t=1}^{N} [y_t - \mu_t(\gamma)]^2 \quad (2.15)$$

where the $y_t$ denotes the $t$-th component of $y$ and $\mu_t(\gamma)$ denotes the $t$-th component of $\mu(\gamma)$. From (2.15) we obtain by a simple calculation:

$$-\frac{\partial \ln p(y|\theta)}{\partial \theta} = \begin{bmatrix} -\frac{1}{\sigma^2} \sum_{t=1}^{N} [y_t - \mu_t(\gamma)] \mu'_t(\gamma) \\ \frac{N}{2\sigma^2} - \frac{1}{\sigma^2} \sum_{t=1}^{N} [y_t - \mu_t(\gamma)]^2 \end{bmatrix} \quad (2.16)$$

where

$$\mu'_t(\gamma) = \frac{\partial \mu_t(\gamma)}{\partial \gamma}. \quad (2.17)$$

Differentiating (2.16) once again gives

$$-\frac{\partial^2 \ln p(y|\theta)}{\partial \theta \partial \theta^T} = \begin{bmatrix} -\frac{1}{\sigma^2} \sum_{t=1}^{N} e_t \mu''_t(\gamma) + \frac{1}{\sigma^2} \sum_{t=1}^{N} \mu'_t(\gamma) \mu''_t(\gamma) & \frac{1}{\sigma^2} \sum_{t=1}^{N} e_t \mu'_t(\gamma) \\ \frac{1}{\sigma^2} \sum_{t=1}^{N} e_t \mu'_t(\gamma) & -\frac{N}{2\sigma^2} + \frac{1}{\sigma^2} \sum_{t=1}^{N} e_t^2 \end{bmatrix} \quad (2.18)$$

where $e_t = y_t - \mu_t(\gamma)$ and

$$\mu''_t(\gamma) = \frac{\partial^2 \mu_t(\gamma)}{\partial \gamma \partial \gamma^T}. \quad (2.19)$$

Taking the expectation of (2.18) and dividing by $N$, we get

$$\frac{1}{N} J = \begin{bmatrix} \frac{1}{\sigma^2} & \frac{1}{\sigma^2} \sum_{t=1}^{N} \mu'_t(\gamma) \mu''_t(\gamma) \\ 0 & \frac{1}{\sigma^2} \end{bmatrix} \begin{bmatrix} 0 \\ \mu''_t(\gamma) \end{bmatrix}. \quad (2.20)$$

We assume that $\mu(\gamma)$ is such that the above matrix has a finite limit as $N \to \infty$. Under this assumption, and the previously-made assumption on $e$, we can also show from (2.18) that

$$-\frac{1}{N} \frac{\partial^2 \ln p(y|\theta)}{\partial \theta \partial \theta^T}$$
2.3 Useful Mathematical Preliminaries and Outlook

converges (as $N \to \infty$) to the right side of (2.20), which concludes the motivation of (2.14). Letting

$$\hat{J} = -\frac{\partial^2 \ln p(y|\theta)}{\partial \theta \partial \theta^T} \bigg|_{\theta = \hat{\theta}}$$

we deduce from (2.14), (2.20), and the consistency of $\hat{\theta}$ that, for sufficiently large values of $N$,

$$\frac{1}{N} \hat{J} \approx \frac{1}{N} J = O(1).$$

Hereafter, $\approx$ denotes an asymptotic (approximate) equality, in which the higher-order terms have been neglected, and $O(1)$ denotes a term that tends to a constant as $N \to \infty$.

Interestingly enough, the assumption that the right side of (2.20) has a finite limit, as $N \to \infty$, holds for many problems, but not for the sinusoidal parameter estimation problem associated with (2.9). In the latter case, (2.22) needs to be modified as follows (see, e.g., [143] and [31]):

$$K_n \hat{J} K_n \approx K_n J K_n = O(1)$$

(2.23)

where

$$K_n = \begin{bmatrix} \frac{1}{N^{1/2}} I_{n_s} & 0 \\ 0 & \frac{1}{N^{1/2}} I_{2n_s+1} \end{bmatrix}$$

(2.24)

and where $I_k$ denotes the $k \times k$ identity matrix; to write (2.24), we assumed that the upper-left $n_s \times n_s$ block of $J$ corresponds to the sinusoidal frequencies, but this fact is not really important for the analysis in this chapter, as we will see later on.

2.3 Useful Mathematical Preliminaries and Outlook

In this section we discuss a number of mathematical tools that will be used in the next sections to derive several important order selection rules. We will keep the discussion at an informal level to make the material as accessible as possible. We first formulate the model order selection as a hypothesis testing problem, with the main goal of showing that the maximum a posteriori (MAP) approach leads to the optimal order selection rule (in a certain sense). Then we discuss the Kullback-Leibler information criterion, which lies at the basis of another approach that can be used to derive model order selection rules.

2.3.1 Maximum a Posteriori Selection Rule

Let $H_n$ denote the hypothesis that the model order is $n$, and let $\overline{n}$ denote a known upper bound on $n$:

$$n \in [1, \overline{n}].$$

(2.25)
We assume that the hypotheses \( \{H_n\}_{n=1}^{\pi} \) are mutually exclusive (i.e., only one of them can hold true at a time). As an example, for a real-valued AR signal with coefficients \( \{a_k\} \) we can define \( H_n \) as follows:

\[
H_n : a_n \neq 0 \text{ and } a_{n+1} = \cdots = a_{\pi} = 0.
\] (2.26)

For a sinusoidal signal we can proceed similarly, after observing that for such a signal the number of components \( n_s \) is related to \( n \) as in (2.11), viz.

\[
n = 3n_s + 1.
\] (2.27)

Hence, for a sinusoidal signal with amplitudes \( \{\alpha_k\} \) we can consider the following hypotheses:

\[
H_{n_s} : \alpha_k \neq 0 \text{ for } k = 1, \ldots, n_s \text{ and } \alpha_k = 0 \text{ for } k = n_s + 1, \ldots, \pi_s
\] (2.28)

for \( n_s \in [1, \pi_s] \) (with the corresponding “model order,” \( n \), being given by (2.27)).

**Remark:** The hypotheses \( \{H_n\} \) can be either nested or non-nested. We say that \( H_1 \) and \( H_2 \) are nested whenever the model corresponding to \( H_1 \) can be obtained as a special case of that associated with \( H_2 \). To give an example, the hypotheses

\[
H_1 : \text{the signal is a first-order AR process}
\]

\[
H_2 : \text{the signal is a second-order AR process}
\]

are nested, whereas the above \( H_1 \) and

\[
H_3 : \text{the signal consists of one sinusoid in noise}
\]

are non-nested.

Let

\[
p_n(y|H_n) = \text{the pdf of } y \text{ under } H_n.
\] (2.29)

Assuming that (2.29) is available, along with the a priori probability of \( H_n \), \( P_n(H_n) \), we can write the conditional probability of \( H_n \), given \( y \), as

\[
P_n(H_n|y) = \frac{p_n(y|H_n)P_n(H_n)}{p(y)}.
\] (2.30)

The maximum a posteriori probability (MAP) rule selects the order \( n \) (or the hypothesis \( H_n \)) that maximizes (2.30). As the denominator in (2.30) does not depend on \( n \), the MAP rule is given by

\[
\max_{n \in [1, \pi]} p_n(y|H_n)P_n(H_n).
\] (2.31)

Most typically, the hypotheses \( \{H_n\} \) are a priori equiprobable, i.e.,

\[
P_n(H_n) = \frac{1}{\pi}, \quad n = 1, \ldots, \pi
\] (2.32)
in which case the MAP rule reduces to (see (2.31))

\[ \max_{n \in [1, \pi]} p_n(y|H_n). \] (2.33)

Next, we define the average (or total) probability of correct detection as

\[ P_{cd} = P\left[ \left( \text{decide } H_1 \right) \cap (H_1 = \text{true}) \right] \cup \cdots \cup \left[ \left( \text{decide } H_\pi \right) \cap (H_\pi = \text{true}) \right]. \] (2.34)

The attribute “average” that has been attached to \( P_{cd} \) is motivated by the fact that (2.34) gives the probability of correct detection “averaged” over all possible hypotheses (as opposed, for example, to only considering the probability of correctly detecting that the model order was 2 (let us say), which is \( P\left( \text{decide } H_2 \right) \cap (H_2 = \text{true}) \)).

**Remark:** Regarding the terminology, note that the determination of a real-valued parameter from the available data is called “estimation,” whereas it is usually called “detection,” or “selection,” for an integer-valued parameter, such as a model order.

In the following we prove that the MAP rule is optimal in the sense of maximizing \( P_{cd} \). To do so, consider a generic rule for selecting \( n \), or, equivalently, for testing the hypotheses \( \{H_n\} \) against each other. Such a rule will implicitly or explicitly partition the observation space, \( \mathbb{R}^N \), into \( \pi \) sets \( \{S_n\}_{n=1}^\pi \), which are such that

we decide \( H_n \) if and only if \( y \in S_n \). (2.35)

Making use of (2.35) along with the fact that the hypotheses \( \{H_n\} \) are mutually exclusive, we can write \( P_{cd} \) in (2.34) as

\[
P_{cd} = \sum_{n=1}^\pi P\left( \left( \text{decide } H_n \right) \cap (H_n = \text{true}) \right)
= \sum_{n=1}^\pi P\left( \left( \text{decide } H_n \right) | H_n \right) P_n(H_n)
= \sum_{n=1}^\pi \int_{S_n} p_n(y|H_n)P_n(H_n) \, dy
= \int_{\mathbb{R}^N} \left[ \sum_{n=1}^\pi I_n(y)p_n(y|H_n)P_n(H_n) \right] \, dy
\] (2.36)

where \( I_n(y) \) is the so-called indicator function given by

\[ I_n(y) = \begin{cases} 
1, & \text{if } y \in S_n \\
0, & \text{otherwise}. 
\end{cases} \] (2.37)

Next, observe that for any given data vector, \( y \), one and only one indicator function can be equal to one (as the sets \( S_n \) do not overlap and their union...
A Review of the AIC, GIC and BIC Rules

is $\mathbb{R}^N$). This observation along with the expression (2.36) for $P_{cd}$ imply that the MAP rule in (2.31) maximizes $P_{cd}$, as stated. Note that the sets $\{S_n\}$ corresponding to the MAP rule are implicitly defined via (2.31); however, $\{S_n\}$ were of no real interest in the proof, as both they and the indicator functions were introduced only to simplify the above proof. For more details on the topic of this subsection we refer to [126; 160].

2.3.2 Kullback-Leibler Information

Let $p_0(y)$ denote the true pdf of the observed data vector, $y$, and let $\hat{p}(y)$ denote the pdf of a generic model of the data. The “discrepancy” between $p_0(y)$ and $\hat{p}(y)$ can be expressed using the Kullback-Leibler (KL) information or discrepancy function (see [80]):

$$D(p_0, \hat{p}) = \int p_0(y) \ln \left( \frac{p_0(y)}{\hat{p}(y)} \right) dy$$

(2.38)

(to simplify the notation, we omit the region of integration when it is the entire space). Letting $E_0\{\cdot\}$ denote the expectation with respect to the true pdf, $p_0(y)$, we can rewrite (2.38) as

$$D(p_0, \hat{p}) = E_0 \left\{ \ln \left( \frac{p_0(y)}{\hat{p}(y)} \right) \right\} = E_0 \{\ln p_0(y)\} - E_0\{\ln \hat{p}(y)\}.$$  

(2.39)

Next, we prove that (2.38) possesses the properties of a suitable discrepancy function, viz.

$$D(p_0, \hat{p}) \geq 0$$

(2.40)

$$D(p_0, \hat{p}) = 0 \text{ if and only if } p_0(y) = \hat{p}(y).$$

To verify (2.40) we use the fact that

$$-\ln \lambda \geq 1 - \lambda \quad \text{for any } \lambda > 0$$

(2.41)

and

$$-\ln \lambda = 1 - \lambda \quad \text{if and only if } \lambda = 1.$$  

(2.42)

Hence, letting $\lambda(y) = \hat{p}(y)/p_0(y)$, we have that

$$D(p_0, \hat{p}) = \int p_0(y) [-\ln \lambda(y)] dy \geq \int p_0(y) [1 - \lambda(y)] dy = \int p_0(y) \left[ 1 - \frac{\hat{p}(y)}{p_0(y)} \right] dy = 0$$

where the equality holds if and only if $\lambda(y) \equiv 1$, i.e. $\hat{p}(y) \equiv p_0(y)$.

The KL discrepancy function can be viewed as showing the “loss of information” induced by the use of $\hat{p}(y)$ in lieu of $p_0(y)$. For this reason, $D(p_0, \hat{p})$ is sometimes called an information function, and the order selection rules derived from it are called information criteria (see the following three sections).
2.3.3 Outlook: Theoretical and Practical Perspectives

Neither the MAP rule nor the KL information can be directly used for order selection because the pdfs of the data vector under the various hypotheses or the true data pdf are usually unknown. A possible way of using the MAP approach for order detection consists of assuming an \textit{a priori} pdf for the unknown parameter vector, $\theta^0$, and integrating $\theta^0$ out of $p_n(y, \theta^0) = p_n(\theta^0)p_n(y|\theta^0)$ (where $p_n(y, \theta^0)$ stands for the joint pdf of $y$ and $\theta^0$) to obtain $p_n(y|H_n)$. This Bayesian-type approach will be discussed later in this chapter. Regarding the KL approach, a natural way of using it for order selection consists of using an estimate, $\hat{D}(p_0, \hat{p})$, in lieu of the unavailable $D(p_0, \hat{p})$ (for a suitably chosen model pdf, $\hat{p}(y)$), and determining the model order by minimizing $\hat{D}(p_0, \hat{p})$. This KL-based approach will be discussed in the next sections.

The derivations of all model order selection rules in the sections that follow rely on the assumption that one of the hypotheses $\{H_n\}$ is true. As this assumption is unlikely to hold in applications with real-life data, the reader may justifiably wonder whether an order selection rule derived under such an assumption has any practical value. To address this concern, let us remark on the fact that good parameter estimation methods (such as the ML method), derived under rather strict modeling assumptions, perform quite well in applications where the assumptions made are rarely satisfied exactly. Similarly, order selection rules based on sound theoretical principles (such as the ML, KL, and MAP) are likely to perform well in applications despite the fact that some of the assumptions made when deriving them do not hold exactly. While the precise behavior of order selection rules (such as those presented in the sections to follow) in various mismodeling scenarios is not well understood, extensive simulation results (see, e.g., [19; 90; 96]) lend support to the above claim.

2.4 Direct Kullback-Leibler (KL) Approach: No-Name Rule

The model-dependent part of the Kullback-Leibler (KL) discrepancy, (2.39), is given by
\[
-E_0\{\ln \hat{p}(y)\}
\] (2.43)
where $\hat{p}(y)$ is the pdf or likelihood of the model (to simplify the notation, we omit the index $n$ of $\hat{p}(y)$; we will reinstate this index later on, when needed). Minimization of (2.43) with respect to the model order is equivalent to maximization of the function
\[
I(p_0, \hat{p}) = E_0\{\ln \hat{p}(y)\}
\] (2.44)
which is sometimes called the relative KL information. The ideal choice for $\hat{p}(y)$ in (2.44) would be the model likelihood, $p_n(y|\theta^0)$. However, the model likelihood function is not available, and hence this choice is not possible. Instead, we might think of using
\[
\hat{p}(y) = p(y|\hat{\theta})
\] (2.45)
in (2.44), which would give

\[ I(p_0, p(y|\hat{\theta})) = E_0\{\ln p(y|\hat{\theta})\}. \]  

(2.46)

Because the true pdf of the data vector is unknown, we cannot evaluate the expectation in (2.46). Apparently, what we could easily do is to use the following unbiased estimate of \( I(p_0, p(y|\hat{\theta})) \), instead of (2.46) itself,

\[ \hat{I} = \ln p(y|\hat{\theta}). \]  

(2.47)

However, the order selection rule that maximizes (2.47) does not have satisfactory properties. This is especially true for nested models, in the case of which the order selection rule based on the maximization of (2.47) fails completely: indeed, for nested models this rule will always choose the maximum possible order, \( \pi \), owing to the fact that \( \ln p_n(y|\hat{\theta}^n) \) monotonically increases with increasing \( n \).

A better idea consists of approximating the unavailable log-pdf of the model, \( \ln p_n(y|\theta^n) \), by a second-order Taylor series expansion around \( \hat{\theta}^n \), and using the so-obtained approximation to define \( \ln \hat{p}_n(y) \) in (2.44):

\[
\ln p_n(y|\theta^n) \approx \ln p_n(y|\hat{\theta}^n) + (\theta^n - \hat{\theta}^n)^T \left[ \frac{\partial \ln p_n(y|\theta^n)}{\partial \theta^n} \bigg|_{\theta^n = \hat{\theta}^n} \right] \\
+ \frac{1}{2} (\theta^n - \hat{\theta}^n)^T \left[ \frac{\partial^2 \ln p_n(y|\theta^n)}{\partial \theta^n \partial \theta^n^T} \bigg|_{\theta^n = \hat{\theta}^n} \right] (\theta^n - \hat{\theta}^n) \equiv \ln \hat{p}_n(y). 
\]  

(2.48)

Because \( \hat{\theta}^n \) is the maximizer of \( \ln p_n(y|\theta^n) \), the second term in (2.48) is equal to zero. Hence, we can write (see also (2.22))

\[ \ln \hat{p}_n(y) \approx \ln p_n(y|\hat{\theta}^n) - \frac{1}{2} (\theta^n - \hat{\theta}^n)^T J (\theta^n - \hat{\theta}^n). \]  

(2.49)

According to (2.12),

\[ E_0((\theta^n - \hat{\theta}^n)^T J (\theta^n - \hat{\theta}^n)) = \text{tr} \left[ J E_0((\theta^n - \hat{\theta}^n)(\theta^n - \hat{\theta}^n)^T) \right] = \text{tr}[I_n] = n \]  

(2.50)

which means that, for the choice of \( \hat{p}_n(y) \) in (2.49), we have

\[ I = E_0 \left\{ \ln p_n(y|\hat{\theta}^n) - \frac{n}{2} \right\}. \]  

(2.51)

An unbiased estimate of the above relative KL information is obviously given by

\[ \ln p_n(y|\hat{\theta}^n) - \frac{n}{2}. \]  

(2.52)

The corresponding order selection rule maximizes (2.52), or, equivalently, minimizes

\[ \text{NN}(n) = -2 \ln p_n(y|\hat{\theta}^n) + n \]  

(2.53)
2.5. Cross-Validatory KL Approach: The AIC Rule

with respect to the model order \( n \). This no-name (NN) rule can be shown to perform better than that based on (2.47), but worse than the rules presented in the next sections. Essentially, the problem with (2.53) is that it tends to overfit (i.e., to select model orders larger than the “true” order). To understand intuitively how this happens, note that the first term in (2.53) decreases with increasing \( n \) (for nested models), whereas the second term increases. Hence, the second term in (2.53) penalizes overfitting; however, it turns out that it does not penalize quite enough. The rules presented in the following sections have a form similar to (2.53), but with a larger penalty term, and they do have better properties than (2.53). Despite this fact, we have chosen to present (2.53) briefly in this section for two reasons: (i) the discussion here has revealed the failure of using \( \max_n \ln p_n(y|\hat{\theta}^n) \) as an order selection rule, and has shown that it is in effect quite easy to obtain rules with better properties; and (ii) this section has laid groundwork for the derivation of better order selection rules based on the KL approach in the next two sections.

To close the present section, we motivate the multiplication with \(-2\) in going from (2.52) to (2.53). The reason for preferring (2.53) to (2.52) is simply due to the fact that for the fairly common NLS model in (2.2) and the associated Gaussian likelihood in (2.3), \(-2\ln p_n(y|\hat{\theta}^n)\) takes on the following convenient form:

\[
-2\ln p_n(y|\hat{\theta}^n) = N \ln \hat{\sigma}_n^2 + \text{constant} \tag{2.54}
\]

(see (2.5)–(2.7)). Hence, in such a case we can replace \(-2\ln p_n(y|\hat{\theta}^n)\) in (2.53) by the scaled logarithm of the residual variance, \( N \ln \hat{\sigma}_n^2 \). This remark also applies to the order selection rules presented in the following sections, which are written in a form similar to (2.53).

2.5 Cross-Validatory KL Approach: The AIC Rule

As explained in the previous section, a possible approach to model order selection consists of minimizing the KL discrepancy between the “true” pdf of the data and the pdf (or likelihood) of the model, or equivalently maximizing the relative KL information (see (2.44)):

\[
I(p_0, \hat{p}) = E_0\{\ln \hat{p}(y)\}. \tag{2.55}
\]

When using this approach, the first (and likely the main) hurdle that we have to overcome is the choice of the model likelihood, \( \hat{p}(y) \). As already explained, ideally we would like to use the true pdf of the model as \( \hat{p}(y) \) in (2.55), i.e. \( \hat{p}(y) = p_n(y|\theta^n) \), but this is not possible since \( p_n(y|\theta^n) \) is unknown. Hence, we have to choose \( \hat{p}(y) \) in a different way. This choice is important, as it eventually determines the model order selection rule that we will obtain.

The other issue we should consider when using the approach based on (2.55) is that the expectation in (2.55) cannot be evaluated because the true pdf of the data is unknown. Consequently, we will have to use an estimate, \( \hat{I} \), in lieu of the unavailable \( I(p_0, \hat{p}) \) in (2.55).
Let \( \mathbf{x} \) denote a fictitious data vector with the same size, \( N \), and the same pdf as \( \mathbf{y} \), but which is independent of \( \mathbf{y} \). Also, let \( \hat{\mathbf{\theta}}_x \) denote the ML estimate of the model parameter vector that would be obtained from \( \mathbf{x} \) if \( \mathbf{x} \) were available (we omit the superindex \( n \) of \( \hat{\mathbf{\theta}}_x \) as often as possible, to simplify notation). In this section we will consider the following choice of the model’s pdf:

\[
\ln \hat{p}(\mathbf{y}) = E_x \{ \ln p(\mathbf{y}|\hat{\mathbf{\theta}}_x) \} \tag{2.56}
\]

which, when inserted in (2.55), yields

\[
I = E_y \{ E_x \{ \ln p(\mathbf{y}|\hat{\mathbf{\theta}}_x) \} \} \tag{2.57}
\]

Hereafter, \( E_x \{ \cdot \} \) and \( E_y \{ \cdot \} \) denote the expectation with respect to the pdf of \( \mathbf{x} \) and \( \mathbf{y} \), respectively. The above choice of \( \hat{p}(\mathbf{y}) \), which was introduced in [4; 6], has an interesting cross-validation interpretation: we use the sample \( \mathbf{x} \) for estimation and the independent sample \( \mathbf{y} \) for validation of the so-obtained model’s pdf. Note that the dependence of (2.57) on the fictitious sample \( \mathbf{x} \) is eliminated (as it should be, since \( \mathbf{x} \) is unavailable) via the expectation operation \( E_x \{ \cdot \} \); see below for details.

An asymptotic second-order Taylor series expansion of \( \ln p(\mathbf{y}|\hat{\mathbf{\theta}}_x) \) around \( \hat{\mathbf{\theta}}_y \), similar to (2.48)–(2.49), yields

\[
\ln p(\mathbf{y}|\hat{\mathbf{\theta}}_x) \approx \ln p(\mathbf{y}|\hat{\mathbf{\theta}}_y) + (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y)^T \left[ \frac{\partial \ln p(\mathbf{y}|\theta)}{\partial \theta} \right]_{\theta = \hat{\mathbf{\theta}}_y} (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y) + \frac{1}{2} (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y)^T \left[ \frac{\partial^2 \ln p(\mathbf{y}|\theta)}{\partial \theta \partial \theta^T} \right]_{\theta = \hat{\mathbf{\theta}}_y} (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y) \tag{2.58}
\]

where \( J_y \) is the \( J \) matrix, as defined in (2.21), associated with the data vector \( \mathbf{y} \). Using the fact that \( \mathbf{x} \) and \( \mathbf{y} \) have the same pdf (which, in particular, implies that \( J_y = J_x \)) along with the fact that they are independent of each other, we can show that

\[
E_y \{ E_x \{ (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y)^T J_y (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y) \} \} = E_y \left\{ E_x \left\{ \operatorname{tr} \left( J_y \left[ (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y - \hat{\mathbf{\theta}}_y)^T \right] \left[ (\hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_x - \hat{\mathbf{\theta}}_y - \hat{\mathbf{\theta}}_y)^T \right] \right) \right\} \right\} = \operatorname{tr} \left[ J_y \left( J_x^{-1} + J_y^{-1} \right) \right] = 2n. \tag{2.59}
\]

Inserting (2.59) in (2.58) yields the following asymptotic approximation of the relative KL information in (2.57):

\[
I \approx E_y \{ \ln p_y(y|\hat{\mathbf{\theta}}^n) - n \} \tag{2.60}
\]

(where we have omitted the subindex \( y \) of \( \hat{\mathbf{\theta}} \) but reinstated the superindex \( n \)). Evidently, (2.60) can be estimated in an unbiased manner by

\[
\ln p_y(y|\hat{\mathbf{\theta}}^n) - n. \tag{2.61}
\]
Maximizing (2.61) with respect to \( n \) is equivalent to minimizing the following function of \( n \):

\[
AIC = -2 \ln p_n(y|\hat{\theta}^n) + 2n
\]

where the acronym AIC stands for Akaike Information Criterion (the reasons for multiplying (2.61) by \(-2\) to get (2.62), and for the use of the word “information” in the name given to (2.62) have been explained before, see the previous two sections).

As an example, for the sinusoidal signal model with \( n_s \) components (see (2.9)), AIC takes on the following form (see (2.6)–(2.11)):

\[
AIC = 2N_s \ln \hat{\sigma}_{n_s}^2 + 2(3n_s + 1)
\]

where \( N_s \) denotes the number of available complex-valued samples, \( \{y_s(t)\}_{t=1}^{N_s} \), and

\[
\hat{\sigma}_{n_s}^2 = \frac{1}{N_s} \sum_{t=1}^{N_s} \left| y_s(t) - \sum_{k=1}^{n_s} \hat{\alpha}_ke^{i(\hat{\omega}_kt + \hat{\phi}_k)} \right|^2.
\]

The performance of AIC has been found to be satisfactory in many case studies and applications to real-life data reported in the literature (see, e.g., [19; 90; 96; 122]). The performance of a model order selection rule, such as AIC, can be measured in different ways, as explained in the next two paragraphs.

As a first possibility, we can consider a scenario in which the data generating mechanism belongs to the class of models under test, and thus there is a true order. In such a case, analytical or numerical studies can be used to determine the probability with which the rule selects the true order. For AIC, it can be shown that, under quite general conditions,

\[
\text{the probability of underfitting} \to 0
\]

\[
\text{the probability of overfitting} \to \text{constant} \geq 0
\]

as \( N \to \infty \) (see, e.g., [72; 96]). We can see from (2.66) that the behavior of AIC with respect to the probability of correct detection is not entirely satisfactory. Interestingly, it is precisely this kind of behavior that appears to make AIC perform satisfactorily with respect to the other possible type of performance measure, as explained below.

An alternative way of measuring the performance is to consider a more practical scenario in which the data generating mechanism is more complex than any of the models under test, which is usually the case in practical applications. In such a case we can use analytical or numerical studies to determine the performance of the model picked by the rule as an approximation of the data generating mechanism: for instance, we can consider the average distance between the estimated and true spectral densities, or the average prediction error of the model. With respect to such a performance measure, AIC performs well, partly because of its tendency to select models with relatively large orders, which may be a good thing to do in a case in which the data generating mechanism is more complex than the models used to fit it.

The non-zero overfitting probability of AIC (see (2.66)) is due to the fact that the term \( 2n \) in (2.62) (that penalizes high-order models), while larger than
the term $n$ that appears in the NN rule, is still “too small”. In effect, extensive simulation studies (see, e.g., [12]) have empirically found that the following Generalized Information Criterion (GIC):

$$\text{GIC} = -2 \ln p_n(y|\hat{\theta}^n) + \nu n$$

(2.67)

may outperform AIC with respect to various performance measures if $\nu > 2$. Specifically, depending on the considered scenario as well as the value of $N$ and the performance measure, values of $\nu$ in the interval $\nu \in [2, 6]$ have been found to give the best performance.

In the next section we show that GIC can be obtained as a natural theoretical extension of AIC. Hence, the use of (2.67) with $\nu > 2$ can be motivated on formal grounds. However, the choice of $\nu$ in GIC is a more difficult problem that cannot be solved in the current KL framework (see the next section for details). The different Bayesian approach, presented later in this chapter, appears to be necessary to arrive at a rule having the form of (2.67) but with a specific expression for $\nu$.

We close this section with a brief discussion on another modification of the AIC rule suggested in the literature (see, e.g., [66]). As explained before, AIC is derived by maximizing an asymptotically unbiased estimate of the relative KL information $I$ in (2.57). Interestingly, for linear regression models (given by (2.2) where $\mu(\gamma)$ is a linear function of $\gamma$), the following corrected AIC rule, $\text{AIC}_c$, can be shown to be an exactly unbiased estimate of $I$:

$$\text{AIC}_c = -2 \ln p_n(y|\hat{\theta}^n) + \frac{2N}{N-n-1}$$

(2.68)

(see, e.g., [20; 66]). As $N \to \infty$, $\text{AIC}_c \to \text{AIC}$ (as expected). However, for finite values of $N$ the penalty term of $\text{AIC}_c$ is larger than that of AIC. Consequently, in finite samples $\text{AIC}_c$ has a smaller risk of overfitting than AIC, and therefore we can say that $\text{AIC}_c$ trades off a decrease of the risk of overfitting (which is rather large for AIC) for an increase in the risk of underfitting (which is quite small for AIC, and hence it can be slightly increased without a significant deterioration of performance). With this fact in mind, $\text{AIC}_c$ can be used as an order selection rule for more general models than just linear regressions, even though its motivation in the general case is pragmatic rather than theoretical. For other finite-sample corrections of AIC we refer the reader to [16; 17; 27].

2.6 Generalized Cross-Validatory KL Approach: The GIC Rule

In the cross-validatory approach of the previous section, the estimation sample $x$ had the same length as the validation sample $y$. In that approach, $\hat{\theta}_x$ (obtained from $x$) was used to approximate the likelihood of the model via $E_x\{p(y|\hat{\theta}_x)\}$. The AIC rule so obtained has a non-zero probability of overfitting (even asymptotically). Intuitively, the risk of overfitting will decrease if
we let the length of the validation sample be (much) larger than that of the estimation sample, i.e.

\[ N \equiv \text{length}(y) = \rho \cdot \text{length}(x), \quad \rho \geq 1. \]  

(2.69)

Indeed, overfitting occurs when the model corresponding to \( \hat{\theta}_x \) also fits the “noise” in the sample \( x \) so that \( p(x|\hat{\theta}_x) \) has a “much” larger value than the true pdf, \( p(x|\theta) \). Such a model may behave reasonably well on a short validation sample \( y \), but not on a long validation sample (in the latter case, \( p(y|\hat{\theta}_x) \) will take on very small values). The simple idea in (2.69) of letting the lengths of the validation and estimation samples be different leads to a natural extension of AIC, as shown below.

A straightforward calculation shows that under (2.69) we have

\[ J_y = \rho J_x \]  

(2.70)

(see, e.g., (2.20)). With this small difference, the calculations in the previous section carry over to the present case and we obtain (see (2.58)–(2.59))

\[
\begin{align*}
I & \approx E_y \{ \ln p(y|\hat{\theta}_y) \} \\
& - \frac{1}{2} E_y \left\{ E_x \left\{ \text{tr} \left( J_y \left[ (\hat{\theta}_x - \theta) - (\hat{\theta}_y - \theta) \right] \left[ (\hat{\theta}_x - \theta) - (\hat{\theta}_y - \theta) \right]^T \right) \right\} \right\} \\
& = E_y \left\{ \ln p(y|\hat{\theta}_y) - \frac{1}{2} \text{tr} \left( J_y \left( \rho J_x^{-1} + J_y^{-1} \right) \right) \right\} \\
& = E_y \left\{ \ln p(y|\hat{\theta}_y) - \frac{1 + \rho}{2} n \right\}. \\
\end{align*}
\]

(2.71)

An unbiased estimate of the right side in (2.71) is given by

\[ \ln p(y|\hat{\theta}_y) - \frac{1 + \rho}{2} n. \]  

(2.72)

The generalized information criterion (GIC) rule maximizes (2.72) or, equivalently, minimizes

\[ \text{GIC} = -2 \ln p_n(y|\hat{\theta}_y^n) + (1 + \rho)n. \]  

(2.73)

As expected, (2.73) reduces to AIC for \( \rho = 1 \). Also, note that, for a given \( y \), the order selected by (2.73) with \( \rho > 1 \) is always smaller than the order selected by AIC (because the penalty term in (2.73) is larger than that in (2.62)); hence, as predicted by the previous intuitive discussion, the risk of overfitting associated with GIC is smaller than for AIC (for \( \rho > 1 \)).

On the negative side, there is no clear guideline for choosing \( \rho \) in (2.73).

As already mentioned in the previous section, the “optimal” value of \( \rho \) in the GIC rule was empirically shown to depend on the performance measure, the number of data samples, and the data generating mechanism itself [12; 96]. Consequently, \( \rho \) should be chosen as a function of all these factors, but, as already stated, there is no clear hint as to how that could be done. The approach of the next section appears to be more successful than the present approach in suggesting a specific choice for \( \rho \) in (2.73). Indeed, as we will see, that approach leads to an order selection rule of the GIC type but with a clear expression for \( \rho \) as a function of \( N \).
2.7 Bayesian Approach: The BIC Rule

The order selection rule to be presented in this section can be obtained in two ways. First, let us consider the KL framework of the previous sections. Therefore, our goal is to maximize the relative KL information (see (2.55)):

\[ I(p_0, \hat{p}) = E_0\{\ln \hat{p}(y)\}. \tag{2.74} \]

The ideal choice of \( \hat{p}(y) \) would be \( \hat{p}(y) = p_n(y|\theta^o) \). However, this choice is not possible since the likelihood of the model, \( p_n(y|\theta^o) \), is not available. Hence we have to use a “surrogate likelihood” in lieu of \( p_n(y|\theta^o) \). Let us assume, as before, that a fictitious sample \( x \) was used to make inferences about \( \theta \). The pdf of the estimate, \( \hat{\theta}_x \), obtained from \( x \) can alternatively be viewed as an a priori pdf of \( \theta \), and hence it will be denoted by \( p(\theta) \) in what follows (once again, we omit the superindex \( n \) of \( \theta, \hat{\theta}, \) etc. to simplify the notation, whenever there is no risk for confusion). Note that we do not constrain \( p(\theta) \) to be Gaussian. We only assume that

\[ p(\theta) \text{ is flat around } \hat{\theta} \tag{2.75} \]

where, as before, \( \hat{\theta} \) denotes the ML estimate of the parameter vector obtained from the available data sample, \( y \). Furthermore, now we assume that the length of the fictitious sample is a constant that does not depend on \( N \), which implies that

\[ p(\theta) \text{ is independent of } N. \tag{2.76} \]

As a consequence of assumption (2.76), the ratio between the lengths of the validation sample and the (fictitious) estimation sample grows without bound as \( N \) increases. According to the discussion in the previous section, this fact should lead to an order selection rule with an asymptotically much larger penalty term than that of AIC or GIC (with \( \rho = \text{constant} \)), and hence with a reduced risk of overfitting.

The scenario introduced above leads naturally to the following choice of surrogate likelihood:

\[ \hat{p}(y) = E_\theta \{p(y|\theta)\} = \int p(y|\theta)p(\theta) \, d\theta. \tag{2.77} \]

To obtain a simple asymptotic approximation of the integral in (2.77) we make use of the asymptotic approximation of \( p(y|\theta) \) given by (2.48)–(2.49)

\[ p(y|\theta) \approx p(y|\hat{\theta})e^{-\frac{1}{2}(\theta-\hat{\theta})^T J(\hat{\theta}-\theta)} \tag{2.78} \]

which holds for \( \theta \) in the vicinity of \( \hat{\theta} \). Inserting (2.78) in (2.77) and using the assumption in (2.75) along with the fact that \( p(y|\theta) \) is asymptotically much
larger at $\theta = \hat{\theta}$ than at any $\theta \neq \hat{\theta}$, we obtain

$$\hat{p}(y) \approx p(y|\hat{\theta}) \frac{p(\hat{\theta})}{|\hat{J}|^{1/2}} \int e^{-\frac{1}{2}(\theta - \hat{\theta})^T \hat{J}^{-1}(\theta - \hat{\theta})} d\theta$$

$$= \frac{p(y|\hat{\theta})p(\hat{\theta})(2\pi)^{n/2}}{|\hat{J}|^{1/2}} \int \frac{1}{(2\pi)^{n/2}|\hat{J}^{-1/2}|} e^{-\frac{1}{2}(\theta - \hat{\theta})^T \hat{J}^{-1/2}(\theta - \hat{\theta})} d\theta$$

$$= \frac{p(y|\hat{\theta})p(\hat{\theta})(2\pi)^{n/2}}{|\hat{J}|^{1/2}}$$

(2.79)

(see [31] and the references therein for the exact conditions under which the above approximation holds true). It follows from (2.74) and (2.79) that

$$\hat{I} = \ln p(y|\hat{\theta}) + \ln p(\hat{\theta}) + \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\hat{J}|$$

(2.80)

is an asymptotically unbiased estimate of the relative KL information. Note, however, that (2.80) depends on the a priori pdf of $\theta$, which has not been specified. To eliminate this dependence, we use the fact that $|\hat{J}|$ increases without bound as $N$ increases. Specifically, in most cases (but not in all; see below) we have that (cf. (2.22))

$$\ln |\hat{J}| = \ln |N \cdot \frac{1}{K} \hat{J}| = n \ln N + \ln \left| \frac{1}{K} \hat{J} \right| = n \ln N + O(1)$$

(2.81)

where we used the fact that $|cJ| = c^n |J|$ for a scalar $c$ and an $n \times n$ matrix $J$. Using (2.81) and the fact that $p(\theta)$ is independent of $N$ (see (2.76)) yields the following asymptotic approximation of the right side in (2.80):

$$\hat{I} \approx \ln p_n(y|\hat{\theta}^n) - \frac{n}{2} \ln N.$$  \hspace{1cm} (2.82)

The Bayesian information criterion (BIC) rule selects the order that maximizes (2.82), or, equivalently, minimizes

$$\text{BIC} = -2 \ln p_n(y|\hat{\theta}^n) + n \ln N.$$  \hspace{1cm} (2.83)

We remind the reader that (2.83) has been derived under the assumption that (2.22) holds, which is not always true. As an example (see [31] for more examples), consider once again the sinusoidal signal model with $n_s$ components, in the case of which we have that (cf. (2.23)–(2.24))

$$\ln |\hat{J}| = \ln |K_n| + \ln |K_n \hat{J} K_n|$$

$$= (2n_s + 1) \ln N_s + 3n_s \ln N_s + O(1)$$

$$= (5n_s + 1) \ln N_s + O(1).$$  \hspace{1cm} (2.84)

Hence, in the case of sinusoidal signals, BIC takes on the form

$$\text{BIC} = -2 \ln p_{n_s}(y|\hat{\theta}^{n_s}) + (5n_s + 1) \ln N_s$$

$$= 2N_s \ln \hat{\sigma}^2_{n_s} + (5n_s + 1) \ln N_s.$$  \hspace{1cm} (2.85)
where $\hat{\sigma}^2_{n_s}$ is as defined in (2.64), and $N_s$ denotes the number of complex-valued data samples.

The attribute Bayesian in the name of the rule in (2.83) or (2.85) is motivated by the use of the a priori pdf, $p(\theta)$, in the rule derivation, which is typical of a Bayesian approach. In fact, the BIC rule can be obtained using a full Bayesian approach, as explained next.

To obtain the BIC rule in a Bayesian framework we assume that the parameter vector $\theta$ is a random variable with a given a priori pdf denoted by $p(\theta)$. Using the product rule (1.6) we can write

$$
p(y|H_n) = \int p_n(y, \theta^n) \, d\theta^n = \int p_n(y|\theta^n)p_n(\theta^n) \, d\theta^n.
$$

(2.86)

The right side of (2.86) is identical to that of (2.77). It follows from this observation and the analysis conducted in the first part of this section that, under the assumptions (2.75) and (2.76) and asymptotically in $N$,

$$
\ln p(y|H_n) \approx \ln p_n(y|\hat{\theta}^n) - \frac{n}{2} \ln N = -\frac{1}{2} \text{BIC}
$$

(2.87)

(see (2.83)). Hence, maximizing $p(y|H_n)$ is asymptotically equivalent with minimizing BIC, independently of the prior $p(\theta)$ (as long as it satisfies (2.75) and (2.76)). The rediscovery of BIC in the above Bayesian framework is important, as it reveals the interesting fact that the BIC rule is asymptotically equivalent to the optimal MAP rule (under the assumption (2.32)), and hence that the BIC rule can be expected to maximize the total probability of correct detection, at least for sufficiently large values of $N$.

The BIC rule has been proposed in [73; 127] among others. In [119; 120] the same rule has been obtained by a rather different approach based on coding arguments and the minimum description length (MDL) principle. The fact that the BIC rule can be derived in several different ways suggests that it may have a fundamental character. In particular, it can be shown that, under the assumption that the data generating mechanism belongs to the model class considered, the BIC rule is consistent; that is,

$$
\text{for BIC: the probability of correct detection} \rightarrow 1 \text{ as } N \rightarrow \infty
$$

(2.88)

(see, e.g., [96; 140]). This should be contrasted with the non-zero overfitting probability of AIC and GIC (with $\rho$ = constant); see (2.65)–(2.66). Note that the result in (2.88) is not surprising in view of the asymptotic equivalence between the BIC rule and the optimal MAP rule.

Finally, we note in passing that if we remove the condition in (2.76) that $p(\theta)$ is independent of $N$, then the term $\ln p(\hat{\theta})$ can no longer be eliminated from (2.80) by letting $N \rightarrow \infty$. Consequently, (2.80) would lead to a prior-dependent rule which could be used to obtain any other rule described in this chapter by suitably choosing the prior. This line of argument can serve the theoretical purpose of interpreting various rules in a Bayesian framework.
2.8 Summary

We begin with the observation that all the order selection rules discussed have a common form, i.e.,
\[-2 \ln p_n(y|\hat{\theta}^n) + \eta(n, N)n\]  
(2.89)
but with different penalty coefficients \(\eta(n, N)\), viz.,
- **AIC**: \(\eta(n, N) = 2\)
- **AIC\(_c\)**: \(\eta(n, N) = \frac{2N}{N - n - 1}\)  
(2.90)
- **GIC**: \(\eta(n, N) = \nu = \rho + 1\)
- **BIC**: \(\eta(n, N) = \ln N\).

Before using any of these rules for order selection in a specific problem, we need to carry out the following steps:

(i) Obtain an explicit expression for the term \(-2 \ln p_n(y|\hat{\theta}^n)\) in (2.89). This requires the specification of the model structures to be tested as well as their postulated likelihoods. An aspect that should receive some attention here is the fact that the derivation of all previous rules assumed real-valued data and parameters. Consequently, complex-valued data and parameters must be converted to real-valued quantities in order to apply the results in this chapter.

(ii) Count the number of unknown (real-valued) parameters in each model structure under consideration. This is easily done in most parametric signal processing problems.

(iii) Verify that the assumptions which have been made to derive the rules hold true. Fortunately, the general assumptions made are quite weak and hence they will usually hold: indeed, the models under test may be either nested or non-nested, and they may even be only approximate descriptions of the data generating mechanism. However, there are two particular assumptions, made on the information matrix \(J\), that do not always hold and hence they must be checked. First, we assumed in all derivations that the inverse matrix, \(J^{-1}\), exists, which is not always the case. Second, we made the assumption that \(J/N = O(1)\). For some models this is not true, and a different normalization of \(J\) is required to make it tend to a constant matrix as \(N \to \infty\) (this aspect is important for the BIC rule only).

We have used the sinusoidal signal model as an example to illustrate the steps above and the involved aspects.

Once the above aspects have been carefully considered, we can go on to use one of the four rules in (2.89)–(2.90) for selecting the order in our estimation problem. The question as to which rule should be used is not an easy one. In general, we can prefer AIC\(_c\) over AIC: indeed, there is empirical evidence that AIC\(_c\) outperforms AIC in small samples (whereas in medium or large samples
the two rules are almost equivalent). We also tend to prefer BIC over AIC or \( \text{AIC}_c \) on the grounds that BIC is an asymptotic approximation of the optimal MAP rule. Regarding GIC, as mentioned earlier in this chapter, GIC with \( \nu \in [2, 6] \) (depending on the scenario under study) can outperform AIC and \( \text{AIC}_c \). Hence, for lack of a more precise guideline, we can think of using GIC with \( \nu = 4 \), the value in the middle of the above interval. In summary, then, a possible ranking of the four rules discussed herein is as follows (the first being considered the best):

- BIC
- GIC with \( \nu = 4 \) (\( \rho = 3 \))
- \( \text{AIC}_c \)
- AIC.

We should warn the reader, however, that the previous ranking is approximate and it will not hold in every application. In Figure 2.1 we show the penalty coefficients of the above rules, as functions of \( N \), to further illustrate the relationship between them.

Finally, we note that in the interest of brevity we will not include numerical examples with the order selection rules under discussion in this chapter, but...
instead refer the reader to later chapters of this thesis as well as to the abundant literature on the subject, see, e.g., [16; 17; 19; 23; 27; 90; 96; 122; 140]. In particular, Chapter 5 contains a host of numerical examples with the information criteria discussed in this review, along with general guidelines as to how a numerical study of an order selection rule should be organized and what performance measures should be used.
Chapter 3

Cross-Validation Rules for Model Order Selection

In this chapter we revisit two cross-validation rules for model order selection which appear to have some advantages over commonly-used rules, yet they do not seem to have received any significant attention so far. In an attempt to make other researchers aware of these rules, we review them here in an informal manner (with most of the technicalities in the original work being omitted) and also compare them with their closest competitors: the AIC and BIC (Akaike and Bayesian Information Criteria) rules.

3.1 Introduction and General Discussion

Most models dealt with in signal processing and elsewhere contain not only real-valued parameters but also integer-valued parameters. Examples of the latter parameters include the order of an AR (Auto-Regressive) equation, the length/order of a FIR (Finite Impulse Response), the number of sinusoidal components in a sinusoids-in-noise signal, the number of sources in an array output equation, the order of a Markov chain, the order of a state-space equation, and so forth. Such integer-valued parameters must be estimated along with the real-valued parameters from the available data. We will use the generic term of “order selection” for the operation of determining the integer-valued parameters of a certain model (even though sometimes “order” may not be a completely appropriate name for the integer-valued parameter in question). Note that some models may contain several integer-valued parameters. However, for the sake of notational simplicity, in what follows we consider the case of a single integer-valued parameter. Nevertheless, everything that we will say...
or any method that we will present can be extended \textit{mutatis mutandis} to the multiple integer-valued parameter case.

Owing to the difficulty of obtaining optimal model selection rules by means of statistical approaches, researchers have considered the more pragmatic approach of developing order selection rules from sound modeling principles such as cross-validation (see, e.g., [153] for a general discussion). An example of a cross-validation-based rule, which we mention in passing but will not consider herein, is the PLS (Predictive Least Squares) rule [121; 164]. Many other such rules exist, based on modeling principles other than cross-validation (for an extensive review on most of the early rules, see [145]). Here we revisit the two cross-validation-based rules for order selection introduced in [145]. While there is no optimality claim attached to these rules, they have a number of desirable properties the most important of which is the fact that these rules make no strong assumption on the data generating mechanism. This fact is explained in the next section where we also clarify several aspects regarding the implementation and properties of these rules and compare them with their closest competitors (and relatives!), viz. the AIC and BIC/MDL rules.

### 3.2 Short-Term and Long-Term Validation Rules

Consider the following linear-regression-like model:

\[
y(t) = \varphi^T(t) \theta_n + \varepsilon(t, \theta_n) \quad t = 1, \ldots, N
\]

where \(\{y(t), \varphi(t)\}_{t=1}^N\) are the available data samples, \(\theta_n\) is the parameter vector, \(n\) is its dimension, and \(\{\varepsilon(t, \theta_n)\}_{t=1}^N\) are the residuals associated with \(\theta_n\). Both \(n\) and \(\theta_n\) are unknown. For a given order \(n\), the perhaps most commonly-used method for estimating \(\theta_n\) is the Least Squares (LS):

\[
\hat{\theta}_n = \arg\min_{\theta_n} V(\theta_n),
\]

\[
V(\theta_n) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta_n).
\]

FIR and AR are two examples of well-known models that can be written in the form of (3.1) with

- **FIR**: \(\varphi^T(t) = [u(t-1) \cdots u(t-n)]\) (\(u(t)\) is the input signal) (3.4)
- **AR**: \(\varphi^T(t) = [-y(t-1) \cdots -y(t-n)]\) (3.5)

\textbf{Remark:} For models with memory, such as the FIR and AR in (3.4) and (3.5) above, the lower index of the sum in (3.3) (and in other similar sums appearing below) should be modified to \(t = n + 1\). Hence, instead of the loss function \(V(\theta_n)\) above we should use the following modified function:

\[
V_n(\theta_n) = \frac{1}{N} \sum_{t=n+1}^N \varepsilon^2(t, \theta_n).
\]
As is well known (and as can be readily checked) \( V(\hat{\theta}_n) \) for \( n = 1, 2, \ldots \) form a non-increasing sequence (if the models are nested), i.e., \( V(\hat{\theta}_n) \geq V(\hat{\theta}_{n+1}) \). A similar property holds for (3.6):

\[
V_n(\hat{\theta}_n) \geq \frac{1}{N} \sum_{t=n+2}^{N} \varepsilon^2(t, \hat{\theta}_n) \geq V_{n+1}(\hat{\theta}_{n+1}).
\] (3.7)

In the following we focus on the relatively simple model class in (3.1) to simplify the discussion as much as possible. However, the extension to a general model class is not complicated and can be found in [145].

Let \( k \) and \( m \) be two integers such that \( km = N \) (for simplicity we assume that \( N \) is exactly divisible by the chosen integers \( k \) and \( m \), which introduces no restriction, see [145]). The choice of \( k \) (or \( m \)) will be discussed later on. Also, let, for \( p = 1, \ldots, k \):

\[
\begin{align*}
I_p &= \{(p-1)m+1, \ldots, pm\} \\
\bar{I}_p &= \{1, \ldots, N\} \setminus I_p \\
\theta_n(p) &= \arg\min_{\theta_n} \sum_{t \in I_p} \varepsilon^2(t, \theta_n) \\
C_{ST}(n) &= \frac{1}{N} \sum_{p=1}^{k} \sum_{t \in \bar{I}_p} \varepsilon^2(t, \hat{\theta}_n(p)) \\
\tilde{\theta}_n(p) &= \arg\min_{\theta_n} \sum_{t \in I_p} \varepsilon^2(t, \theta_n) \\
C_{LT}(n) &= \frac{1}{(k-1)N} \sum_{p=1}^{k} \sum_{t \in \bar{I}_p} \varepsilon^2(t, \tilde{\theta}_n(p)).
\end{align*}
\] (3.8)

In the criterion \( C_{ST}(n) \) above, \( \bar{\theta}_n(p) \) is obtained from all subsamples except the \( p \)-th one, and the corresponding squared residuals for the remaining \( p \)-th subsample are summed-up (over \( p \)) to get the criterion. Hence, \( C_{ST}(n) \) is a short-term (ST) cross-validation rule whose estimation subsample is \((k-1)\) times larger than the validation subsample. For \( C_{LT}(n) \), on the other hand, the estimate \( \tilde{\theta}_n(p) \) is obtained from the \( p \)-th subsample only, and its corresponding squared residuals for the remaining \((k-1)\) subsamples (that were not used for estimation) are summed-up (over \( p \)) to obtain the criterion. Consequently \( C_{LT}(n) \) is a long term (LT) cross-validation rule for which the length of the estimation subsample is \((k-1)\) times smaller than that of the validation subsample.

With the above interpretation of \( C_{ST}(n) \) and \( C_{LT}(n) \) in mind, a sound rule for selecting \( n \) might consist of minimizing \( C_{ST}(n) \) and \( C_{LT}(n) \) over a given set of orders. Intuitively the order estimate obtained from \( C_{ST}(n) \) should be larger than that derived from \( C_{LT}(n) \) (because high-order models are better short-term than long-term extrapolators). The minimization of \( C_{ST}(n) \) or \( C_{LT}(n) \), however, would be computationally rather expensive. This computational difficulty issue was addressed in [145] where the following much simpler asymptotic approximations of \( C_{ST}(n) \) and \( C_{LT}(n) \) were derived:
\[ \hat{C}_{ST}(n) = V(\hat{\theta}_n) + \frac{2}{N} \sum_{p=1}^{k} w_p^T \hat{R}^{-1} w_p \]  
\[ \text{and} \]
\[ \hat{C}_{LT}(n) = V(\hat{\theta}_n) + \frac{k}{N} \sum_{p=1}^{k} w_p^T \hat{R}^{-1} w_p. \]  

In the above equations \( \hat{\theta}_n \) is the LS estimate as defined in (3.2),

\[ \hat{\theta}_n = \left[ \sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \left[ \sum_{t=1}^{N} \varphi(t)y(t) \right] \triangleq \hat{R}^{-1}\hat{r}, \]  

\[ V(\hat{\theta}_n) \] is the corresponding LS criterion,

\[ V(\hat{\theta}_n) = \frac{1}{N} \sum_{t=1}^{N} y^2(t) - \frac{\hat{\varphi}^T \hat{R}^{-1} \hat{r}}{N} \]  

and

\[ w_p = \sum_{t \in I_p} \varphi(t)\epsilon(t, \hat{\theta}_n) = \sum_{t \in I_p} \left\{ \varphi(t)y(t) - \left[ \varphi(t)\varphi^T(t) \right] \hat{\theta}_n \right\}. \]  

As already indicated in the remark following equation (3.5), for models with memory (such as (3.4) and (3.5)) the lower limit of the sums in (3.13) and (3.14) should be replaced by \( t = n + 1 \). On the other hand, when computing the sums over \( I_p \) in (3.15) we can use the last \( n \) samples from \( I_{p-1} \) (except for \( I_1 \)); proceeding in this way we can obtain these sums from the corresponding terms of \( \hat{r} \) and \( \hat{R} \) without any extra computation.

Let \( \nu = [n_{\min}, n_{\max}] \) be the interval to which the sought order is assumed to belong. Then we can estimate the order \( n \) by one of the following two rules:

**Short Term Validation (STV) Rule:**

\[ \hat{n}_{STV} = \arg \min_{n \in \nu} \hat{C}_{ST}(n). \]  

**Long Term Validation (LTV) Rule:**

\[ \hat{n}_{LTV} = \arg \min_{n \in \nu} \hat{C}_{LT}(n). \]  

A number of comments on the STV and LTV rules follow.
3.2. Short-Term and Long-Term Validation Rules

C1. The criterion \( \hat{C}_{LT}(n) \) is a good approximation of \( C_{LT}(n) \) iff both \( k \) and \( m \) are “sufficiently large”, whereas \( \hat{C}_{ST}(n) \) only requires that \( k \) be “large” (\( m \) can even be equal to one), see \[145\]. However, intuitively it is quite clear that \( \hat{C}_{ST}(n) \) with a small value of \( m \), such as \( m = 1 \), may have a tendency to overestimate the “true order” (because high-order models often are good very short-term (e.g. one-step) extrapolators, as already mentioned). Consequently, in general we may also want to use \( \hat{C}_{ST}(n) \), like \( \hat{C}_{LT}(n) \), with an \( m \gg 1 \) (even though this is not required for \( \hat{C}_{ST}(n) \) to be a good approximation of \( C_{ST}(n) \)). The choice of \( m \) (or \( k \)) is further discussed in the comment C3 below.

C2. Neither \( \hat{C}_{ST}(n) \) nor \( \hat{C}_{LT}(n) \) imposes any restriction on the “true system” that generated the data \( \{y(t), \varphi(t)\} \). In particular, it is not required that the “true system” belongs to the model set in (3.1). This fact appears to be an advantage over the many existing order selection rules that were derived under the following assumption:

\[
\text{The data was generated by Equation (3.1) with } n = n^* (\text{the “true order”) and } \theta_{n^*} = \theta_n^* (\text{the “true parameter vector”), and moreover } \{\varepsilon(t, \theta_n^*)\} \text{ is a white (usually Gaussian) noise sequence.} \tag{3.18}
\]

Clearly (3.18) is a strong assumption. Interestingly, it can be shown \[145\] that under (3.18) and for \( N \gg 1 \) the STV rule is equivalent to the AIC rule \[3\]:

\[
\hat{n}_{AIC} = \arg \min_{n \in \nu} \left[ \ln V(\hat{\theta}_n) + \frac{2n}{N} \right], \tag{3.19}
\]

whereas the LTV rule is equivalent to a form of the BIC/MDL rule (\[5; 119; 127\]) given by

\[
\hat{n}_{BIC} = \arg \min_{n \in \nu} \left[ \ln V(\hat{\theta}_n) + \frac{nk}{N} \right] \tag{3.20}
\]

(for \( m = 1 \) the equivalence between STV and AIC holds under more general conditions than (3.18), see \[145\]). Practical data, of course, do not satisfy the assumption (3.18) without which neither AIC nor BIC/MDL can be given the intuitively appealing cross-validation interpretation afforded by the aforementioned equivalence with the STV and LTV rules.

C3. To apply the STV or LTV rules we must choose \( k \) (or \( m \)). Inspired by the equivalence in (3.20) with a BIC-like rule and the fact that BIC uses \( k = \ln N \) (see Chapter 2), as well as for the lack of a better choice, we may select, for both STV and LTV:

\[
k = \left\lfloor c \ln N \right\rfloor \tag{3.21}
\]

where \( \lfloor x \rfloor \) denotes the integer part of \( x \) and \( c \) is a constant (such as \( c = 1.5 \), see later). Note that for any practical value of \( N \) and \( c \) we have \( k > 2 \). Then if both rules are used with the same value of \( k \) (and if the model set considered has a nested structure in the sense that a model corresponding to a given order
can be obtained as a special case of a model with a larger order, which is true for (3.1)) the following inequality holds (see Appendix B for a proof):

\[ \hat{n}_{\text{STV}} \geq \hat{n}_{\text{LTV}}. \] (3.22)

Note that for the AIC and the BIC/MDL rules a similar inequality holds, viz. \( \hat{n}_{\text{AIC}} \geq \hat{n}_{\text{BIC}} \), even if the model set is not nested.

C4. The cross-validation rules STV and LTV are more demanding from a computational standpoint than AIC or BIC, but only slightly. Indeed, because \( \hat{R} \) and \( \hat{r} \) (as well as their subsample counterparts, see the discussion following equation (3.15)) are available from the computation of \( \hat{\theta}_n \), the number of additional flops required by STV or LTV, for a given value of \( n \), is of order \( O(kn^2) \), which typically is a small fraction of the number of flops required to compute AIC or BIC.

C5. For \( n \) larger than the “true order” \( \varepsilon(t, \hat{\theta}_n) \) will be almost independent of \( n \) (at least for high SNRs and \( N \gg n \)) and hence \( \varepsilon_n \) will have a nested structure as a function of \( n \). Because \( \hat{R} \) always has such a nested structure it can readily be checked that in such a case the second term in both (3.11) and (3.12) increases with increasing \( n \). This behavior is similar to that of AIC or BIC and it is desirable since it reduces the risk of overestimation. For \( n \) less than the “true order”, on the other hand, the second term in (3.11) and (3.12) may in principle also decrease as \( n \) increases. This possible behavior of the STV and LTV criteria, which cannot occur for AIC or BIC, is also desirable because it may reduce the risk of underestimation. Note that for \( n \) less than the “true order”, the assumption (3.18) is invalid and hence the equivalence between the STV and LTV rules, on the one hand, and AIC and BIC, on the other, does not hold. For such order values the former rules may thus behave quite differently from the latter.

3.3 Numerical Examples

First we consider an example in which the data is generated by a fourth-order AR equation:

\[ y(t) - 1.1y(t - 1) + 1.2y(t - 2) - 0.9y(t - 3) + 0.7y(t - 4) = \epsilon(t) \] (3.23)

where \( \epsilon(t) \) is Gaussian white noise with zero mean and unit variance. The model class is given by (3.1) and (3.5).

An obvious way of testing the quality of a model order selection rule in this case, in which the true data-generating process is included in the model set, is to consider the percentage of correctly selected order. If, however, the intended use of the model is taken into consideration, then other criteria, such as the prediction ability of the models selected by the rule, may be of interest. The optimal one-step predictor for the AR process in (3.1) and (3.5) is

\[ \hat{y}(t) = -a_1y(t - 1) - a_2y(t - 2) \cdots - a_ny(t - n), \] (3.24)
where \([a_1, a_2, \ldots, a_n]^T\) is the parameter vector \(\theta_n\) in (3.1). For an \(m\)-step prediction, (3.24) is simply used iteratively with the true values, when unavailable, replaced with their estimates (see, e.g., [146]).

We may also want to use the estimated models for spectral analysis, and with this goal in mind we can compare the mean estimated power spectral densities (PSDs) as well as the root mean square errors (RMSEs) of the PSD estimates. The PSD of an ARMA process is given by [147]

\[
\Phi(\omega) = \sigma^2 \frac{|B(e^{i\omega})|^2}{|A(e^{i\omega})|^2}
\]  

(3.25)

where \(B(\cdot)\) and \(A(\cdot)\) are the MA and AR polynomials, and \(\omega \in [0, 2\pi]\) is the frequency variable. We use (3.25) to calculate the PSDs of the true processes (AR and ARMA, see below) as well as of the estimated AR models.

All examples given in this section were run for several values of \(N\) but we present results only for \(N = 240\), for conciseness reasons. For the STV and LTV rules, a choice of the integer parameter \(k\) was needed. Based on our empirical experience and (3.21) we selected \(k = \lfloor 1.5 \ln N \rfloor\). This value of \(k\) was used in all the examples presented here (\([1.5 \ln 240] = 8\)). For all presented plots, 1000 different realizations of the true process were used and the model order selection rules were tried on AR models with orders in the interval \(\nu = [1, 40]\) (\(\nu = [2, 40]\) for the sinusoidal process below). For the prediction plots, one prediction was made for each of the 1000 realizations. Then, using the true data generating process, 1000 different realizations of the true process were obtained for each prediction. The standard deviations of the predicted values from these “true” values were calculated for all 1000 predictions.

In Figure 3.1, the histograms of the orders selected by the different rules are shown. The bars at \(n = 4\) represent the frequency of the correctly selected order. Observe that the LTV and BIC rules behave similarly, but the BIC rule has a slightly higher frequency of correct order selection. The STV and AIC both select very high orders. In Figure 3.2, the standard deviations of the prediction errors for different prediction steps are shown. As expected, the standard deviations converge to a value which is equal to the standard deviation of the true process (as the predicted values converge to zero). Also note that the LTV and BIC rules offer almost identical performance and that they are superior to the STV and AIC. Figure 3.3 shows the mean PSDs of the models with orders given by the different rules together with the true PSD of the process. Figure 3.4 shows the RMSEs of the PSD estimates. Once again, the LTV and BIC rules perform very similarly, and clearly better than the STV and AIC which both give spurious peaks and large RMSEs.

Next we consider processes which are not included in the model set. First we do this by generating data using the ARMA equation:

\[
y(t) - 1.4y(t - 1) + 1.5y(t - 2) - 0.8y(t - 3) + 0.2y(t - 4) =
\]

\[
e(t) - 1.2e(t - 1) + 1.1e(t - 2) - 0.9e(t - 3) + 0.8e(t - 4).
\]  

(3.26)

We use the same performance criteria as for the AR process, and the results are shown in Figures 3.5-3.8. Note that there is no longer any “true” order
for the rules to select. However, the histogram of the selected orders may still be interesting and is therefore included. The performances of the LTV and BIC are again similar and better (in terms of prediction and PSD estimation quality) than the performances of the STV and AIC rules (note, however, that the STV and AIC rules provide better models for estimating the PSD in the valley regions).

Finally, we consider data generated by the following sinusoids-in-noise process:

\[ y(t) = 0.2 \sin(0.3\pi t + \phi_1) + \sin(0.51\pi t + \phi_2) + \sin(0.53\pi t + \phi_3) + e(t) \]  

where \( \{e(t)\} \) is a zero mean Gaussian white noise with variance \( \sigma^2 \) and the initial phases \( \{\phi_i\} \) are uniformly distributed in the interval \([0, 2\pi]\). Two values of \( \sigma^2 \) were considered giving the following signal-to-noise ratios (SNR = \(1/\sigma^2\)):

\[
\begin{align*}
\text{SNR} = 10 \text{ dB} & \quad \Rightarrow \quad \sigma^2 = 0.10 \\
\text{SNR} = 0 \text{ dB} & \quad \Rightarrow \quad \sigma^2 = 1.0.
\end{align*}
\]

For each SNR value we present the same plots as for the previous examples (see Figures 3.9-3.14), except that we omit the RMSEs of the PSD estimates that cannot be calculated for this process since the power spectrum of the sinusoids is infinite at the frequencies of the sinusoids. Also note that in the present case the AR models obtained will have some poles very close to the unit circle, and hence the prediction will converge much slower than before.

The following observations can be made by inspecting the Figures 3.9-3.14. Both the prediction and PSD estimation performance of the models provided by the STV and AIC rules are better in this case than the corresponding performance of the LTV and BIC-based models. This is at least true if the prediction horizon is not too long (the very long-term predictions given by the STV and AIC models tend to be unstable, see Figures 3.10 and 3.13), and if the few spurious peaks in the PSD spectra of the STV and AIC models (see especially Figure 3.11) are ignored. Regarding the ranking of the LTV and BIC rule, they behave quite similarly for SNR = 0 dB, but for SNR = 10 dB the LTV rule outperforms BIC with respect to both prediction and PSD estimation performance.

### 3.4 Concluding Remarks

It is hard to make decisive claims about the order selection rules under discussion based on a rather limited experience. Hence we will not impose any “conclusions” on the reader, but make only some suggestions. The STV and AIC rules appear to behave in almost the same fashion. Their performance for AR and ARMA processes (the model was always AR) was rather unreliable in our experiments. On the other hand, they had a better overall performance than the LTV and BIC rules for sinusoids-in-noise processes. Regarding the latter two rules, the LTV had a slight advantage over the BIC in some cases but usually their behaviors were very similar to one another.
3.4. Concluding Remarks

Figure 3.1: Histogram of the selected orders for the different model order selection rules. The data was given by the AR(4) process in (3.23).

On a different note, the STV and LTV rules were derived virtually without making any assumption on the data generating mechanism. While this fact should give them an edge over the AIC and BIC rules (usually obtained by making rather restrictive assumptions on the process that generated the data), the latter rules do not appear to have any particular problem when used on data that cannot be exactly fitted by the assumed model.

Therefore, to conclude, which rule do we recommend? The unreliability of STV and AIC in some cases was disconcerting. Consequently we tend to suggest LTV or BIC, perhaps used with a lower value of \( k \) when high order models are expected based on the \( a \) priori information on the data at hand.
3. Cross-Validation Rules for Model Order Selection

Figure 3.2: Standard deviations of the AR prediction errors for different prediction steps. The data was given by the AR(4) process in (3.23).

Figure 3.3: Mean PSDs of the AR models. The data was given by the AR(4) process in (3.23).
Figure 3.4: RMSEs of the PSDs of the AR models. The data was given by the AR(4) process in (3.23).

Figure 3.5: Histogram of the selected orders for the different model order selection rules. The data was given by the ARMA(4,4) process in (3.26).
Figure 3.6: Standard deviations of the AR prediction errors for different prediction steps. The data was given by the ARMA(4,4) process in (3.26).

Figure 3.7: Mean PSDs of the AR models. The data was given by the ARMA(4,4) process in (3.26).
3.4. Concluding Remarks

Figure 3.8: RMSEs of the PSDs of the AR models. The data was given by the ARMA(4,4) process in (3.26).

Figure 3.9: Histogram of the selected orders for the different model order selection rules. The data was given by the sinusoids-in-noise process in (3.27) with SNR = 10 dB.
Figure 3.10: Standard deviations of the AR prediction errors for different prediction steps. The data was given by the sinusoids-in-noise process in (3.27) with SNR = 10 dB.
Figure 3.11: Mean PSDs of the AR models together with the true line spectrum of the sinusoids. The data was given by the sinusoids-in-noise process in (3.27) with SNR = 10 dB. The image in the upper left corner is a zoom around the two closely spaced peaks.
Figure 3.12: Histogram of the selected orders for the different model order selection rules. The data was given by the sinusoids-in-noise process in (3.27) with SNR = 0 dB.

Figure 3.13: Standard deviations of the AR prediction errors for different prediction steps. The data was given by the sinusoids-in-noise process in (3.27) with SNR = 0 dB.
Figure 3.14: Mean PSDs of the AR models together with the true line spectrum of the sinusoids. The data was given by the sinusoids-in-noise process in (3.27) with SNR = 0 dB.
3. Cross-Validation Rules for Model Order Selection
Chapter 4

On Information Criteria and the Generalized Likelihood Ratio Test of Model Selection

The Information Criterion (IC) rule and the Generalized Likelihood Ratio Test (GLRT) have been usually considered to be two rather different approaches to model order selection. However, we show here that a natural implementation of the GLRT is in fact equivalent to the IC rule. A consequence of this equivalence is that a specific IC rule, such as AIC or BIC, can be viewed as a more direct way of implementing a GLRT with a specific threshold. Another consequence of the equivalence, which is emphasized herein, is a possibly original way of exploiting the information provided by the local behavior of an IC for selecting the structure of sparse models (the parameter vectors of which comprise “many” elements equal to zero).
4.1 Introduction: The IC and the GLRT

Our discussion of the IC rule and the GLRT will be relatively brief. Let

\[ y \in \mathbb{R}^N \]  
\[ \theta^n \in \mathbb{R}^n \]  
\[ p(y|\theta^n) = \text{the likelihood function corresponding to the model of order } n \text{ with parameters } \theta^n \]  
\[ \hat{\theta}^n = \arg \max_{\theta^n} p(y|\theta^n) \]  
\[ [n_l, n_u] \]  

\[ \hat{\theta}^n \]  
\[ [n_l, n_u] \]  

The Generalized IC (GIC) rule selects the order \( n \) as the solution to the following minimization problem:

\[ \min_{n \in [n_l, n_u]} \left[ \text{GIC}_n \triangleq -2 \ln p(y|\hat{\theta}^n) + \eta \cdot n \right]. \]  

(4.1)

The penalty factor \( \eta \) has different expressions for different IC rules. In particular, for the Akaike IC (AIC),

\[ \eta_{\text{AIC}} = 2 \]  

(4.2)

whereas for the Bayesian IC (BIC),

\[ \eta_{\text{BIC}} = \ln N \]  

under some “regularity conditions” (4.3)

(see, e.g., Chapter 2).

The GLRT is based on the following distributional result. Let

\[ \text{GLRT}_{m,p} \triangleq \left[ -2 \ln p(y|\hat{\theta}^m) \right] - \left[ -2 \ln p(y|\hat{\theta}^p) \right] \text{ for } p > m. \]  

(4.4)

Then (see, e.g., [53; 76]):

\[ \text{GLRT}_{m,p} \sim \chi^2(p - m) \quad \text{(for } N \gg 1, \text{ and } p > m \geq \text{true order}) \]  

(4.5)

where \( \chi^2(p - m) \) denotes the chi-square distribution with \( (p - m) \) degrees of freedom. Based on (4.5), we can decide between two models with orders \( m \) and \( p \), respectively, as follows:

\[ \text{GLRT}_{m,p} \overset{m}{\underset{p}{\leq}} \eta(p - m) \]  

(4.6)

where \( \eta(p - m) \) is a threshold that depends on the difference \( (p - m) \); we may choose \( \eta(p - m) \) based on the (asymptotic) distributional result in (4.5), or possibly in a different manner (see below).

The GLRT can be used to select an order \( n \in [n_l, n_u] \) in several ways. In the next section we show that a natural implementation of the GLRT for order selection is equivalent to the GIC rule. The connection between the GLRT and the GIC for comparing \( \text{two model orders} \), let us say \( m \) and \( p \), is evident...
from the definitions in (4.1) and (4.4), which are both based on $-2 \ln p(y|\hat{\theta}^m)$ and $-2 \ln p(y|\hat{\theta}^p)$, see [141] for details. However, here we are interested in the equivalence between the GLRT and GIC rules applied to all model orders in the interval $[n_l, n_u]$. Regarding this type of equivalence, it was observed in [111] that the order selected by GIC is the same as the order selected by GLRT applied to all possible model pairs, provided that the thresholds in the GLRT are chosen in a certain way (which was not given explicitly in [111]). The equivalence between the much more efficient implementation of GLRT presented here and GIC, however, appears to be a novel result, or at least a result that is not well known. In effect, in the previous literature authors who supported AIC or BIC usually criticized GLRT (see, e.g., [19]), and vice-versa (see, e.g., [54]).

### 4.2 The Main Result: The Equivalence Between GIC and GLRT

A “typical” variation of $-2 \ln p(y|\hat{\theta}^n)$, as $n$ increases, is shown in Figure 4.1. Let

$$n_1 < n_2 < \cdots < n_s \quad \text{for some } s \leq n_u - n_l$$

(4.7)

denote the values of $n$ for which the decrease of $-2 \ln p(y|\hat{\theta}^n)$ to the next order ($n + 1$) is “insignificant”, whereas the decrease from the previous order ($n - 1$) is “significant”; see the figure. Mathematically, $n_1, \ldots, n_s$ are the values of $n \in [n_l, n_u]$ for which the following inequalities are satisfied:

$$\begin{cases} 
\text{GLRT}_{n,n+1} < \eta \\
\text{GLRT}_{n-1,n} > \eta 
\end{cases}$$

(4.8a)

(4.8b)

where $\eta$ is a short notation for the $\eta(1)$ in (4.6). Note that for $n = n_l$ only (4.8a) is required, whereas for $n = n_u$ only (4.8b) should hold. According to (4.6), $\{n_1, \ldots, n_s\}$ are possible candidates for a GLRT-based selected order.

The next step compares the candidate models with orders $n_1, \ldots, n_s$, obtained in the previous step, with one another. Similarly to (4.8), this can be done by selecting the values of $n$ in the set $\{n_1, \ldots, n_s\}$ for which

$$\begin{cases} 
\text{GLRT}_{n_k,n_{k+1}} < (n_{k+1} - n_k)\eta \\
\text{GLRT}_{n_{k-1},n_k} > (n_k - n_{k-1})\eta 
\end{cases}$$

(4.9a)

(4.9b)

(once again, for $n = n_1$, only (4.9a) is required, whereas for $n = n_s$ only (4.9b) is necessary). Observe that the basic threshold $\eta$ in (4.8) has been multiplied by $(n_{k+1} - n_k)$ and $(n_k - n_{k-1})$ in (4.9); this multiplication is suggested by the number of degrees of freedom in the distributional result (4.5).

Let $\{\bar{n}_1, \ldots, \bar{n}_s\}$ be the values of $n$ obtained in the step above. By definition, we have $\bar{s} < s$. We continue in the same fashion until we end-up with a single value of $n$, which is the order selected by GLRT.

Next, we note from the definitions in (4.1) and (4.4) that

$$\text{GIC}_m - \text{GIC}_p = \text{GLRT}_{m,p} - (p - m)\eta$$

(4.10)
for any positive integers $p$ and $m$, with $p > m$. Assuming that the $\eta$ in (4.10) (and hence in (4.1)) is the same as the $\eta$ in (4.8) and (4.9), it follows from (4.10) that the inequalities in (4.9) (and therefore in (4.8), which is a special case of (4.9)) are equivalent to

$$
\begin{aligned}
\{ & \text{GIC}_{n_k} < \text{GIC}_{n_{k+1}} \\
& \text{GIC}_{n_k} < \text{GIC}_{n_{k-1}}. \}
\end{aligned}
$$

This observation allows us to deduce that: the first step of the GLRT outlined above finds all local minima of GIC$_n$ for $n \in [n_l, n_u]$; the next step of GLRT finds all local minima of the sequence of local minimum values obtained in the first step; and so on until the last step of GLRT selects the value of $n$ that is the global minimum of \{GIC$_n$\} for $n \in [n_l, n_u]$.

The equivalence between the previous implementation of GLRT and the GIC rule is thus established. As a consequence of this equivalence, the GIC rule can be viewed as a more direct way of implementing the above multi-step GLRT procedure. Furthermore, any specific GIC rule, such as AIC and BIC, can be seen as an elaborated way of choosing the threshold in the GLRT. The above equivalence also suggests a possibly original way of using GIC to select the structure of sparse models, which is described in the next section.

### 4.3 A Corollary: Using the GIC Rule for Sparse Models

Let $\theta^n_k$ denote the $k$-th element of $\theta^n$. The parameter vector associated with a sparse model, by definition, comprises “many” elements equal to zero. For a sparse model, we may thus be interested not only in selecting the order $n$, but also in deciding which of the elements of the parameter vector are zero and which ones are different from zero. This means that, for such a model, we can define a structure selection problem (which includes the order selection) as follows: decide which elements $\{\theta^n_k\}$ of the parameter vector corresponding to the model of maximum order $n_u$ are different from zero. This problem could in principle be solved by using the GIC rule applied to the models whose parameter vectors contain all possible combinations of zero and non-zero elements. However, even for small values of $n_u$ the number of combinations to be tested is considerable, and the required calculations get quickly out of hand. A much simpler way of using GIC for structure selection of sparse models is suggested by the equivalence to GLRT, as explained next.

Consider, once again, the plot of $[-2 \ln p(y|\theta^n)]$ versus $n$ in Figure 4.1. This kind of variation may well occur when the data were generated by a sparse model. To see this, we first remind the reader that each decrease of $-2$ times the log-likelihood function which is greater than the threshold $\eta$ is “significant”. Consequently, if we let $\hat{n}_C$ denote the order selected by the GLRT outlined in the previous section, and let $\hat{n}_1, \ldots, \hat{n}_C$ denote the orders smaller or equal to $\hat{n}_C$ for which the decrease from the previous order is larger than $\eta$, we can decide that

$$
\theta^n_{\hat{n}_1} \neq 0, \ldots, \theta^n_{\hat{n}_C} \neq 0.
$$

(4.12)
On the other hand, since any decrease of $-2$ times the log-likelihood function which is smaller than $\eta$ is “insignificant”, we can infer that

$$\theta_{k}^{n_u} = 0 \quad \text{for} \quad k \in \{n_1, \ldots, n_u \} \setminus \{\tilde{n}_1, \ldots, \tilde{n}_G \}. \tag{4.13}$$

To transfer these arguments to GIC we invoke the equivalence shown in the previous section, according to which $\tilde{n}_G$ is the global minimum of $\{\text{GIC}_n\}$ and $\tilde{n}_1, \ldots, \tilde{n}_G$ are the orders smaller than or equal to $\tilde{n}_G$ for which $\text{GIC}_n$ decreases from the previous order. Then, based on (4.12) and (4.13), we can decide that

$$\begin{cases} 
\theta_{k}^{n_u} \neq 0 & \text{for} \quad k \in \{\tilde{n}_1, \ldots, \tilde{n}_G \} \\
\theta_{k}^{n_u} = 0 & \text{else.} \tag{4.14}
\end{cases}$$

The solution in (4.14) to the structure selection problem, obtained by means of GIC, can be refined, if desired, by using GIC to compare it with structures obtained from it by “slight perturbations”: in this way the number of structures to be tested is kept reasonably small (anyway, much smaller than the number of all possible structures). An indication of whether the structure corresponding to (4.14) is a reasonable choice can be obtained as follows. Let $\hat{\theta}^{\tilde{\alpha}_G}$ denote the ML estimate of $\theta^{\tilde{\alpha}_G}$ obtained by enforcing the zero element constraints given by (4.14); and let $\text{GIC}_{\tilde{n}_G}$ denote the value of GIC associated with $\hat{\theta}^{\tilde{\alpha}_G}$. If

$$\text{GIC}_{\tilde{n}_G} < \text{GIC}_{\tilde{\alpha}_G} \tag{4.15}$$

then we can infer that the sparse structure in (4.14) is a good choice (note that $\text{GIC}_{\tilde{n}_G}$ is obtained from (4.1) using the number of non-zero elements in $\hat{\theta}^{\tilde{\alpha}_G}$ as the order $n$).

To illustrate the main points made in the discussion above, we present the results of a Monte-Carlo simulation study in the remainder of this section.

### 4.3.1 Numerical Example

The data are generated by means of the equation:

$$z(t) = h_1 u(t - 1) + \cdots + h_{15} u(t - 15) + e(t), \quad t = 1, \ldots, N \tag{4.16}$$

where $\{z(t)\}$ denotes the output sequence, $\{u(t)\}$ is a known, white Gaussian input with mean zero and variance $\lambda^2 = 1$, $\{e(t)\}$ is a white Gaussian noise with mean zero and variance $\sigma^2 = 1$, and $h_3 = h_4 = h_6 = h_7 = h_8 = h_9 = h_{11} = h_{12} = h_{13} = 0$. \tag{4.17}

The other coefficients of the sparse FIR channel in (4.16), namely $\{h_1, h_2, h_5, h_{10}, h_{14}, \text{and} \ h_{15}\}$ are first independently drawn from a uniform distribution on $[-1, -0.2] \cup [0.2, 1]$. We then normalize $\{h_k\}$ so that $\sum_{k=1}^{15} h_k^2 = 10$; consequently, the signal-to-noise ratio (SNR) for (4.16) is equal to

$$\text{SNR} = 10 \log_{10} \left[ \lambda^2 \left( \sum_{k=1}^{15} h_k^2 \right) / \sigma^2 \right] = 10 \text{ [dB]}.$$
We use data sequences of two lengths: \( N = 200 \) and \( N = 500 \). We make use of \( 10^6 \) realizations of the data sequence, obtained by varying the noise sequence, the input sequence, and the non-zero FIR coefficients. For each data realization we use the BIC rule, see (4.1) and (4.3), and the procedure outlined in the first part of this section, see (4.14), to select the model structure. Note that for (4.16) with \( \bar{n} \) tap coefficients \( \{h_1, \ldots, h_{\bar{n}}\} \) we have \( \theta^n = [\sigma^2, h_1, \ldots, h_{\bar{n}}]^T \) and hence \( n = \dim(\theta^n) = \bar{n} + 1 \). We assume that we know that \( \bar{n} \leq 20 \), and hence set \( n_u = 21 \). Consequently, to select the model structure by using (4.14) we compute \( \text{BIC}_n \) for \( n \in [1, 21] \).

Figure 4.2 shows the average results obtained from the ensemble of realizations mentioned above. As can be seen, the performance of the simple solution in (4.14) to the structure selection problem is quite satisfactory. Moreover, we note that for the structure selected with (4.14) the inequality in (4.15) was satisfied in about 87% (for \( N = 200 \)) and 96% (for \( N = 500 \)) of the total number of cases considered; in other words, the sparse structure selected was preferable (according to BIC) to any other full structure with \( \bar{n} \in [0, 20] \) in 87% and 96% of the cases respectively.
4.3. A Corollary: Using the GIC Rule for Sparse Models

Figure 4.1: A "typical" variation of $[-2\ln p(y|\hat{\theta}^n)]$ for increasing values of $n \in [n_l, n_u]$. 
Figure 4.2: The histogram (averaged over different channel realizations) of the decision event that $h_k \neq 0$, as a function of $k$; the stars indicate the ideal result.
Chapter 5

Multi-Model Approach to Model Selection

The single-model approach to model selection based on information criteria, such as AIC or BIC, is omnipresent in the signal processing literature. However, any single-model approach picks up only one model and hence misses the potentially significant information associated with the other models fitted to the data. In our opinion this is a drawback: indeed, depending on the application, even the true model structure (assuming that there was one) may not be the best choice for the intended use of the model. The multi-model approach does not suffer from such a problem: using nothing more than the values of AIC or BIC it estimates the a posteriori probabilities of each model under consideration and then it goes on to use all fitted models in a weighted manner. We show via a numerical study that the multi-model approach can outperform the single-model approach in terms of statistical accuracy, without unduly increasing the computational burden. The first goal of this chapter is to advocate the multi-model approach. A second goal is to suggest some guidelines for numerically studying the performance of a model selection rule.

5.1 Introduction and Preliminaries

Almost any model used in parametric (or model-based) signal processing applications comprises both integer-valued and real-valued parameters that must be estimated from the available data. The integer-valued parameters of a model are usually referred to by the generic term of “orders”. Examples of such parameters include the degree of a polynomial trend, the orders of an autore-
gressive moving-average equation, the dimension of a state-space equation, the number of sinusoidal components in a noisy sinusoidal signal, the number of source signals impinging on an array of sensors, and the number of coefficients in the impulse response of a linear channel.

Two well-known rules for order selection are the Akaike Information Criterion (AIC) [4] and the Bayesian Information Criterion (BIC) [73; 119; 127]; see also [19; 23; 90; 96; 140] and Chapter 2 of this thesis. These order selection rules will also be in the focus of the present chapter. Both AIC and BIC are tied to the use of the maximum likelihood (ML) method, or an asymptotic approximation thereof, for estimating the real-valued parameters of the considered models. While the ML method is often used in signal processing applications, it is not always the method of choice and hence the dependence of AIC and BIC on the ML may seem to be a limitation. However, this is not necessarily so, as we explain next. Let

\[ y \in \mathbb{R}^N = \text{the data vector} \]
\[ \theta^n \in \mathbb{R}^n = \text{the parameter vector of the model of order } n \]
\[ p(y|\theta^n) = \text{the data likelihood associated with the model of order } n \text{ with parameters } \theta^n \]
\[ \hat{\theta}^n = \text{arg max} p(y|\theta^n) = \text{the ML estimate of } \theta^n \]
\[ [n_l, n_u] = \text{a known interval for the model order } n \text{ (the indices } l \text{ and } u \text{ indicate the lower and upper limits, respectively).} \]

The AIC and BIC rules for model order selection are based on \( \{p(y|\theta^n)\} \) for \( n \in [n_l, n_u] \). However, let us assume that \( \{p(y|\hat{\theta}^n)\} \) are not available. This is the case, for instance, when instead of the ML estimates \( \{\hat{\theta}^n\} \) we obtain parameter estimates, let us say \( \{\hat{\theta}^n\} \), by a different (usually simpler) method. While in such a case AIC and BIC are not directly applicable, we can derive similar order selection rules based on \( \{p(\hat{\theta}^n|\theta^n)\} \) instead of \( \{p(y|\hat{\theta}^n)\} \), assuming that the distribution function \( p(\hat{\theta}^n|\theta^n) \) has a known functional form (which is often the case, at least for \( N \gg 1 \)); see [70] for this variation on the AIC and BIC theme that makes the information criterion approach to order selection more widely applicable.

In the signal processing literature AIC and BIC are predominantly used in a single-model selection fashion. However, the selection of a single model is often unsatisfactory since it ignores the potentially significant information associated with (some of) the other models fitted to the data. To emphasize this fact, we give examples of situations in which even selecting the true-order model is not the best choice for the intended use of the estimated model! First, consider a prediction application based on fitting a polynomial trend to \( N = 20 \) noisy data samples generated by a polynomial of order \( n_0 = 10 \). Owing to the noise present in the data, estimating 10 parameters from only 20 data samples will generally result in poor estimation accuracy and hence poor prediction performance; presumably, a polynomial trend based on an underestimated order \( n < n_0 \) will perform better. Next, assume data consisting of \( n_0 = 2 \) closely-spaced sinusoidal signals, with frequencies \( \omega_0 \) and \( \omega_0 + \Delta \), in strong (possibly
colored) noise. In such a case many sinusoidal parameter estimation methods used with \( n = 2 \) will not resolve the two sinusoidal components; instead, they will provide a frequency estimate in the vicinity of \( \omega_0 \) and another spurious estimate corresponding to a “noise spectral peak”. However, if we apply such a method with an overestimated order \( n > n_0 \), then the two closely-spaced sinusoidal signals may be resolved and their parameters well estimated.

The multi-model approach to model selection does not suffer from the above pitfalls of the single-model approach. The multi-model approach associates posterior probabilities with each estimated model of order \( n \in [n_l, n_u] \) and uses all of these models in a weighted manner according to their estimated posterior likelihoods; see, e.g., [19; 64; 74; 110; 163]. The present chapter is an attempt to advocate the multi-model approach for signal processing applications.

### 5.2 AIC and BIC Rules for Single-Model Selection

The AIC rule is usually derived using the Kullback-Leibler information [4], whereas the BIC rule is obtained as an asymptotic (in \( N \)) estimate of the a posteriori probability of the model given the data [73; 127]. However, both criteria can also be obtained in a common cross-validation framework as briefly explained below; see Chapter 2 and the references therein for more details.

Let \( x \) denote a fictitious data sample with the same size and the same distribution function as \( y \), but independent of \( y \); and let \( \hat{\theta}^n \) denote the ML estimate of \( \theta^n \) that would be obtained from \( x \) if \( x \) were available. The log-likelihood of the model of order \( n \) can then be assessed in the following cross-validatory manner:

\[
E_{\hat{\theta}^n}[\ln p(y|\hat{\theta}^n)] (5.1)
\]

where \( x \) serves as the “estimation sample” and \( y \) as the “validation sample”, and where the expectation with respect to \( \hat{\theta}^n \) was taken to eliminate the dependence on a specific (and, anyway, unavailable) sample \( x \). As is known, the reason for using cross-validation is that the in-sample likelihood \( p(y|\hat{\theta}^n) \) is an overestimate of the “true” likelihood (owing to “noise fitting”), whereas the cross-validatory likelihood \( p(y|\tilde{\theta}^n) \) is a more accurate estimate. After some simple asymptotic approximations as well as taking a partial expectation with respect to \( y \) (see, e.g., Chapter 2) we obtain the following estimate for (5.1):

\[
\ln p(y|\tilde{\theta}^n) - n. (5.2)
\]

This leads to the AIC rule:

\[
\min_{n \in [n_l, n_u]} \left[ \text{AIC}_n = -2 \ln p(y|\tilde{\theta}^n) + 2n \right] (5.3)
\]

which is obviously equivalent to maximizing (5.2) with respect to \( n \).

The BIC rule can be obtained in the same cross-validation framework by making the assumption that the fictitious estimation sample \( x \) has now a constant length (independent of \( N \)), and using the following cross-validatory expression for the log-likelihood of the model of order \( n \), slightly modified from
(5.1) \[
\ln \left\{ E_{\hat{\theta}^n} [p(y|\hat{\theta}^n)] \right\}.
\] (5.4) 

A derivation similar to that which leads to AIC can be used to show that the minimization of \(-2\) times an asymptotic approximation of (5.4) yields the BIC rule:

\[
\min_{n \in [n_l, n_u]} \left[ \text{BIC}_n = -2 \ln p(y|\hat{\theta}^n) + (\ln N)n \right].
\] (5.5)

Note that we can rewrite (5.4) in a Bayesian framework:

\[
\ln \left( \int p(y|\theta^n)p(\theta^n)d\theta^n \right) = \ln p(y|M_n)
\] (5.6)

where \(M_n\) denotes the hypothesis that the model of order \(n\) has generated the data, and where \(p(\theta^n)\) can now be interpreted as the \textit{a priori} distribution function of \(\theta^n\). Since (5.6) equals (5.4) and since (5.5) is \(-2\) times an asymptotic approximation of (5.4) (and therefore of (5.6)) it follows that \(\text{BIC}_n\) is an asymptotic approximation (or estimate) of \(-2\ln p(y|M_n)\). From the discussion in the first part of this section it can be seen that \(\text{AIC}_n\) can be given a similar interpretation, with the difference that the prior distribution function of \(\theta^n\) corresponding to \(\text{AIC}_n\) is dependent on \(N\).

In the literature, the performance of AIC, BIC and other order selection rules is often discussed in terms of their probability of true-order selection. With respect to this measure of performance, BIC is superior to AIC, at least for \(N \gg 1\). However, the probability of true-order selection is not the indicator of performance we are necessarily interested in: as explained in the introduction, in applications the true-order model may perform worse than other estimated models. A numerical study of the performance of AIC and BIC based on the intended use of the model will be reported in the last section of this chapter.

5.3 AIC and BIC Rules for Multi-Model Selection

The following discussion concentrates on the BIC rule but its conclusions also apply to the AIC more or less directly. As explained in the previous section (see the discussion around (5.6)), for \(N \gg 1\),

\[
\text{BIC}_n \approx -2 \ln p(y|M_n) + \text{const}
\] (5.7)

or equivalently

\[
p(y|M_n) \approx \text{const} \cdot e^{-\frac{1}{2} \text{BIC}_n}.
\] (5.8)

Under the assumption that all orders \(n \in [n_l, n_u]\) are \textit{a priori} equiprobable (i.e. \(P(M_n) = \text{const}\)), we derive from (5.8) that

\[
P(M_n|y) = \frac{P(y|M_n)P(M_n)}{p(y)} \approx \text{const} \cdot e^{-\frac{1}{2} \text{BIC}_n}.
\] (5.9)
5.3. AIC and BIC Rules for Multi-Model Selection

The normalizing constant in (5.9) can be obtained from the condition that

$$
\sum_{k=n_l}^{n_u} P(M_k|y) \approx \text{const} \cdot \sum_{k=n_l}^{n_u} e^{-\frac{1}{2}BIC_k} = 1.
$$

(5.10)

Inserting the constant given by (5.10) into (5.9) yields

$$
\hat{P}(M_n|y) = \frac{e^{-\frac{1}{2}BIC_n}}{\sum_{k=n_l}^{n_u} e^{-\frac{1}{2}BIC_k}},
$$

(5.11)

which are the estimated \textit{a posteriori} probabilities of the hypothesized model structures \(\{M_n\}_{n=n_l}^{n_u}\) provided by BIC.

The BIC rule for single-model selection picks up the \textit{mode} of \(\{\hat{P}(M_n|y)\}\). Similarly, we can think of selecting the \textit{median} or the \textit{mean} of \(\{\hat{P}(M_n|y)\}\). While comparing these possible variations of BIC with one another may be an interesting task, as explained in the introductory section we do not recommend a single-model approach for model selection. Instead, we suggest using all estimated models corresponding to \(\{\hat{\Theta}_n\}_{n=n_l}^{n_u}\) along with their associated estimated posterior probabilities \(\{\hat{P}(M_n|y)\}\), for \(n = n_l, \ldots, n_u\). Presumably such a multi-model approach should be preferable to the single-model approach, especially when \(\hat{P}(M_n|y)\) is relatively flat around its maximum with respect to \(n\).

Results on, or usages of, the multi-model approach to model selection are scarce in the signal processing literature; a notable exception is [137] where a multi-model-based prediction approach was proposed. On the other hand, in the statistical literature the multi-model approach is catching up with the single-model approach (see, e.g., [19; 64; 74; 110; 163]). In the latter literature the following slight variation of the multi-model approach is sometimes recommended. Let \(n_1, \ldots, n_L\) be the values of \(n\) in the interval \([n_l, n_u]\) for which

$$
\hat{P}(M_n|y) \geq \max_{n \in [n_l, n_u]} \hat{P}(M_n|y) \quad \text{for } (\text{e.g., } \rho = 10).
$$

(5.12)

Then, instead of (5.11), we can use the following truncated (and re-normalized) posterior likelihood estimates of \(\{M_n\}\):

$$
\hat{P}(M_n|y) = \begin{cases} 
\frac{e^{-\frac{1}{2}BIC_n}}{\sum_{k=1}^{L} e^{-\frac{1}{2}BIC_k}} & \text{for } n = n_1, \ldots, n_L \\
0 & \text{else.}
\end{cases}
$$

(5.13)

However, if the interval \([n_l, n_u]\) was judiciously chosen then the condition in (5.12) will be satisfied for many or all of the values of \(n\) under test, and hence (5.13) will yield similar results to those obtained by (5.11). Additionally, the choice of \(\rho\) in (5.12) is necessarily subjective.\(^1\) With these facts in mind we will not consider the modification in (5.13) in what follows.

\(^1\)We note in passing that (5.13) with \(\rho = 1\) gives the single-model BIC rule!
Next, let us assume that we intend to use the data model for prediction. Let \( z \) denote the variable (scalar or vector valued) we wish to predict using the available data in \( y \). The conditional mean of \( z \) given \( y \) is well known to yield the optimal (in the mean square sense) prediction:

\[
\tilde{z} = E(z|y) = \int z p(z|y) dz
\]  

(5.14)

where the conditional probability distribution function can be written as

\[
p(z|y) = \sum_{n=n_l}^{n_u} \int p(z|\theta^n, M_n, y) d\theta^n
\]

(5.15)

Assuming that \( p(\theta^n|M_n, y) \) is well approximated by a Dirac impulse centered at \( \theta^n = \hat{\theta}^n \) (which is true for \( N \gg 1 \)), the integral in (5.15) can be shown to be approximately equal to \( p(z|\hat{\theta}^n, M_n, y) \). Using this approximation in (5.14) and (5.15) yields:

\[
\tilde{z} \approx \sum_{n=n_l}^{n_u} P(M_n|y) \int z p(z|\hat{\theta}^n, M_n, y) dz = \sum_{n=n_l}^{n_u} P(M_n|y) \hat{z}_n
\]  

(5.16)

where \( \hat{z}_n \) is the optimal prediction obtained using the estimated model of order \( n \) (as if it were the true data generating mechanism). By replacing the unknown posterior probabilities \( P(M_n|y) \) in (5.16) by their estimates \( \hat{P}(M_n|y) \) we finally obtain a multi-model based prediction of \( z \) in the form of a weighted average:

\[
\tilde{z} = \sum_{n=n_l}^{n_u} \hat{P}(M_n|y) \hat{z}_n.
\]  

(5.17)

While \( \tilde{z} \) in (5.17) is only an asymptotically valid approximation of the optimal prediction \( \tilde{z} \), as explained above, (5.17) has an appealing form because it requires only standard calculations to obtain \( \hat{z}_n \).

For other applications than prediction we should proceed in a slightly different manner. In particular, we should not apply directly the weighted-average type of formula in (5.17) to the quantity of interest, such as a spectral density, since using the multi-model approach in that way is not guaranteed to yield better performance than a single-model approach. Instead we can make use of the multi-model approach as follows. Assume that the models \( \{M_n\} \) are nested such that, for any \( \bar{n} > n \), the model \( M_{\bar{n}} \) with the parameter vector equal to \( \begin{bmatrix} \theta^n \\ 0 \end{bmatrix} \) reduces to the model \( M_n \) with parameter vector \( \theta^n \). In particular this means that the \( n_u \times 1 \) vectors \( \begin{bmatrix} \hat{\theta}^n \\ 0 \end{bmatrix} \) for \( n \in [n_l, n_u] \) can all be viewed as estimates of \( \theta^n \). Similarly to the equation (5.14) for prediction, the optimal (in
the mean square sense) estimate of $\theta^{n_u}$ is given by the mean of the a posteriori distribution $p(\theta^{n_u}|y)$:

$$
\hat{\theta}^{n_u} = \int \theta^{n_u} p(\theta^{n_u}|y) d\theta^{n_u}
= \int \theta^{n_u} \left[ \sum_{n=n_0}^{n_u} p(\theta^{n_u}, M_n|y) \right] d\theta^{n_u}
= \sum_{n=n_0}^{n_u} P(M_n|y) \int \theta^{n_u} p(\theta^{n_u}|M_n, y) d\theta^{n_u}
\approx \sum_{n=n_0}^{n_u} P(M_n|y) \left[ \hat{\theta}^n_0 \right]
$$

(5.18)

where, like in (5.16), the approximate equality is motivated by the fact that, for $N \gg 1$, the function $p(\theta^{n_u}|M_n, y)$ is well approximated by a Dirac impulse centered at $\theta^{n_u} = \left[ \hat{\theta}^n_0 \right]$. Replacing $\{P(M_n|y)\}$ in (5.18) by their estimates $\{\hat{P}(M_n|y)\}$ provided by BIC (or AIC) yields the following asymptotically optimal estimate of $\theta^{n_u}$:

$$
\hat{\theta}^{n_u} = \hat{P}(M_{n_1}|y) \left[ \hat{\theta}^{n_1}_0 \right] + \hat{P}(M_{n_2}|y) \left[ \hat{\theta}^{n_2+1}_0 \right] + \cdots + \hat{P}(M_{n_u}|y) \hat{\theta}^{n_u}_0.
$$

(5.19)

The use of $M_{n_u}$ with the multi-model based parameter estimate $\hat{\theta}^{n_u}$ should outperform (for $N \gg 1$) the use of $M_{n_u}$ with any single-model based parameter estimate $\left[ \hat{\theta}^n_0 \right]$ ($n \leq n_u$), or equivalently, the use of $M_n$ with any $\hat{\theta}^n$, for any $n \in [n_l, n_u]$.

**Remark:** Note that the use of $M_{n_u}$ with $\hat{\theta}^{n_u}$ for prediction leads to (5.17) whenever the predictors are linear in the model parameters. However, when the predictor is a nonlinear function of the model parameters, the use of (5.17) and (5.19) may yield different prediction results in finite samples, but the differences between these results should be small for $N \gg 1$.

### 5.4 Numerical Study of Performance

#### 5.4.1 Performance Measures

Most studies of model selection rules reported in the literature make use of data generated using only one model $M_{n_0} \in \{M_n\}$, in a multiple realization manner. A possible concern with this type of approach is the fact that the data generating mechanism is assumed to belong to the class of models considered. However this is not the concern we want to discuss here. In our opinion a major concern with the aforementioned numerical approach to performance study is the use of only one (or a few) data generating mechanism(s). This is not satisfactory in several respects. First, if we choose $n_0$ closer to $n_l$ than to $n_u$ then we favor the selection rules that underestimate the order; and vice-versa if $n_0$ is chosen closer to $n_u$ than to $n_l$. Hence we should generate data...
sets using all order values \( n \) in \([n_l, n_u]\). Second, using only one data generating mechanism, for each order \( n \) may lead to biased conclusions. Consequently, for each order \( n \) in \([n_l, n_u]\) we should use as many data generating mechanisms as possible.

Note that the idea of considering all values of \( n \) in \([n_l, n_u]\) in a performance study is also supported by the classical hypothesis testing theory where the performance and optimality of a rule are defined with respect to the total (or average) probability of correct selection, which using our notation is given by (see, e.g., [126; 160] and Chapter 2 in this thesis)

\[
P(\text{decide } M_n | M_n) P(M_n) + \cdots + P(\text{decide } M_{n_u} | M_{n_u}) P(M_{n_u}).
\] (5.20)

If the model structures considered are \( a \ priori \) equiprobable, as assumed before, then (5.20) reduces to

\[
\frac{1}{n_u - n_l + 1} \sum_{n=n_l}^{n_u} P(\text{decide } M_n | M_n).
\] (5.21)

The point to note is that it is the total probability (5.20), or the average probability (5.21), which is of interest, and not \( P(\text{decide } M_n | M_n) P(M_n) \) for only one value of the order in the assumed interval \([n_l, n_u]\).

In summary, we should generate the data as follows: given \( N, n_u, \) and \( n_l, \) for each \( n \) in \([n_l, n_u]\) draw \( n_m \) model parameter vectors \( \{\theta^n\} \) from a random distribution, and for each \( \theta^n \) obtain via replication \( n_r \) random data realizations of length \( N \) (by varying the “noise” part of the data generating mechanism). In this way we obtain

\[(n_u - n_l + 1) \times n_m \times n_r \] data sets

and we should determine the average performance of the order selection rule under analysis over the ensemble of these data sets.

The “average performance” mentioned in the previous paragraph should be defined with respect to the objective of the model. For instance, if the model is to be used for prediction then we can utilize the average prediction mean square error (MSE):

\[
\frac{1}{(n_u - n_l + 1)n_m n_r} \sum_{n} \sum_{m} \sum_{r} (\hat{z} - z)^2_{n,m,r}
\] (5.23)

where \( n_m \) and \( n_r \) have been defined before, and where \( [\hat{z} - z]^2_{n,m,r} \) denotes the element-wise squared prediction error corresponding to the \( r \)-th realization \((r = 1, \ldots, n_r)\) of the data set generated by the \( m \)-th mechanism \((m = 1, \ldots, n_m)\) of order \( n \) \((n = n_l, \ldots, n_u)\). Whenever the objective of the model is not yet determined (or perhaps not well defined) or when the model order is a quantity of interest \( per \ se \) we can use the average frequency of various selected orders as a possible performance measure (see, however, the introductory section for a cautionary note about using this measure):

\[
\frac{1}{(n_u - n_l + 1)n_m n_r} \sum_{n} \sum_{m} \sum_{r} \delta_{n+k,\hat{n}_{n,m,r}} \quad \text{for } k = 0, \pm 1, \pm 2, \ldots
\] (5.24)
where \( \hat{n}_{n,m,r} \) denotes the order estimated from the data set corresponding to \((n, m, r)\), and \( \delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \).

### 5.4.2 Numerical Examples

We consider a linear channel described by the equation

\[
y(k) = \theta_0 u(k) + \cdots + \theta_{n-1} u(k - n + 1) + e(k)
\]  

(5.25)

where \( \{e(k)\} \) is a Gaussian white noise and \( \{u(k)\} \) is a known white Gaussian training sequence. The coefficients \( \{\theta_k\} \) are unknown and so is the order \( n \) of the channel. For the finite-impulse-response model in (5.25), a simple calculation shows that (see, e.g., [19; 23; 90; 96; 140])

\[
-2 \ln p(y|\hat{\theta}^n) = N \ln \hat{\sigma}_n^2 + \text{const}
\]

(5.26)

where \( \hat{\theta}^n = [\hat{\theta}_0 \cdots \hat{\theta}_{n-1}]^T \),

\[
\hat{\sigma}_n^2 = \frac{1}{N} \sum_{k=1}^{N} [y(k) - \varphi^T(k)\hat{\theta}^n]^2,
\]

and where

\[
\varphi^T(k) = [u(k) \cdots u(k-n+1)]
\]

\[
\hat{\theta}^n = \left[ \sum_{k=1}^{N} \varphi(k) \varphi^T(k) \right]^{-1} \left[ \sum_{k=1}^{N} \varphi(k) y(k) \right].
\]

Furthermore for (5.25) the prediction of \( \{y(k)\}_{k=N+1,N+2,\ldots} \) based on the estimated model of order \( n \) (assumed to be the true data generating mechanism) is given by

\[
\hat{y}(k) = \hat{\theta}_0 u(k) + \cdots + \hat{\theta}_{n-1} u(k - n + 1), \quad k = N + 1, N + 2, \ldots
\]

We consider the following values of \( N, n_l, n_u, n_m \) and \( n_r \):

\[
\begin{align*}
N &= 100 \quad \text{and} \quad N = 400 \\
n_l &= 1 \quad n_u = 10 \\
n_m &= 100 \quad n_r = 100.
\end{align*}
\]

(5.27)

For each data model, the \( n_r = 100 \) noise and training sequences are drawn from two independent Gaussian distributions with means equal to zero and variances equal to 1 (for the training sequence) and to the values corresponding to the SNRs below (for the noise); here SNR denotes the signal-to-noise ratio, for which we consider the following two values:

\[
\text{SNR} = 0 \text{ dB and SNR} = 10 \text{ dB}.
\]

(5.28)

The coefficients \( \{\theta_k\}_{k=0}^{n-1} \) in the \( n_m = 100 \) sets of coefficients for each order \( n \in \left[n_l, n_u\right] \) are independently drawn from a uniform distribution in the interval
Figure 5.1: Histograms of estimated orders and averaged posterior probabilities for $N = 100$ and SNR = 0 dB.
Figure 5.2: Histograms of estimated orders and averaged posterior probabilities for $N = 400$ and SNR = 0 dB.
Figure 5.3: Histograms of estimated orders and averaged posterior probabilities for \( N = 100 \) and \( \text{SNR} = 10 \) dB.
Figure 5.4: Histograms of estimated orders and averaged posterior probabilities for $N = 400$ and SNR = 10 dB.
[-1, 1]. To maintain a constant SNR, see (5.28), we normalize each vector \( \theta^n = [\theta_0 \cdots \theta_{n-1}]^T \) generated as above, so that \( \|\theta^n\| = 1 \).

Figures 5.1-5.4 show the histograms of the orders selected by AIC and BIC for the four combinations of \( N \) and SNR in (5.27) and (5.28); see (5.24) for the definition of the histogram. We also show the average posterior probability estimates for the considered orders, obtained using AIC and BIC; see (5.11) for BIC; the corresponding formula for AIC is similar. We can see from Figures 5.1-5.4 that, as expected, AIC overestimates the true order more often than BIC, and that BIC underestimates the true order more often than AIC; also BIC selects the true order most often. Note that, among other things, the histograms of estimated orders are useful in evaluating the Cramér-Rao bounds and the estimation accuracy of the parameter estimates when not only the real-valued parameters but also the orders are estimated, see, e.g., [125].

Figures 5.5-5.8 show the corresponding average prediction MSE curves (see (5.23) for a definition). The differences between the performances of the various model selection rules are small, which appears to be due to the fact that in the present case the true order has a much higher posterior likelihood than the other orders, as indicated by the BIC approximation in Figures 5.1-5.4). However, despite being relatively small, the differences between the various model selection rules are consistent (the curves in Figures 5.5-5.8 do not intersect). Note from the figures that, as expected, the multi-model approach outperforms

Figure 5.5: Average prediction MSEs for \( N = 100 \) and SNR = 0 dB.
Figure 5.6: Average prediction MSEs for $N = 400$ and SNR = 0 dB.
Figure 5.7: Average prediction MSEs for $N = 100$ and SNR = 10 dB.
Figure 5.8: Average prediction MSEs for $N = 400$ and SNR = 10 dB.
the single-model approach.

5.5 Conclusions

In this chapter we have studied the multi-model approach. We have shown how well-known model selection criteria, such as AIC and BIC, can be used to obtain estimates of the \textit{a posteriori} probabilities of the considered models. These probabilities can be used, e.g., to average predictions from several models (5.17), or to obtain an averaged estimate of the model parameters (5.19). We have shown, by means of numerical examples, that the multi-model approach can offer better prediction performance than the single-model approach based on AIC or BIC.
Chapter 6

Adaptive Equalization for Frequency-Selective Channels of Unknown Length

This chapter studies adaptive equalization for time-dispersive communication channels whose impulse responses have unknown lengths. This problem is important, because an adaptive equalizer designed for an incorrect channel length is suboptimal.

We propose a novel, systematic approach to the problem under study, which circumvents the estimation of the channel length. The key idea is to model the channel impulse response via a mixture Gaussian model, which has one component for each possible channel length. The parameters of the mixture model are estimated from a received pilot sequence. We derive the optimal receiver associated with this mixture model, along with some computationally efficient approximations of it. Via numerical simulations, we show that our method can outperform conventional adaptive Viterbi equalizers that use a fixed or an estimated channel length.

6.1 Introduction

6.1.1 Problem Formulation and Background

We consider single-carrier transmission over a frequency-selective channel. Suppose that one transmits a sequence of complex symbols \( \{ s(n) \}^{N+N_{\text{post}}-1}_{n=-N_{\text{pre}}} \) (where each symbol is selected from a finite alphabet), composed of an \( N_{\text{pre}} \) symbols
long preamble \( \{s(n)\}_{n=-N_{\text{pre}}}^{N-1} \) (which is known to the receiver), an \( N \)-symbol data sequence \( \{s(n)\}_{n=0}^{N-1} \) (unknown to the receiver), and an \( N_{\text{post}} \) symbols long postamble \( \{s(n)\}_{n=N+N_{\text{post}}}^{N+L-1} \) (known to the receiver) in a standard fashion.\(^1\) The received data then take on the form

\[
y(n) = \sum_{l=0}^{L} h_l s(n - l) + e(n), \quad n = 0, \ldots, N + L - 1
\]  

(6.1)

where \( \{h_l\}_{l=0}^{L} \) is the symbol-sampled baseband impulse response of the channel, and \( L \) is its length.\(^2\) Also, in (6.1), \( e(n) \) is a noise term. Note that the samples received before \( n = 0 \) and after \( n = N + L - 1 \) do not depend on the data symbols, so if \( \{h_l\} \) is known and \( e(n) \) is white, then \( \{y(n)\}_{n=0}^{N+L-1} \) is a sufficient statistic for the detection of \( \{s(n)\}_{n=0}^{N-1} \). If the channel \( \{h_l\} \) and the properties of \( e(n) \) are known, then the transmitted data \( \{s(n)\}_{n=0}^{N-1} \) can be recovered from \( \{y(n)\}_{n=0}^{N+L-1} \) by using standard equalization techniques [97; 114]. One of the most well known equalizers is the maximum-likelihood sequence detector (MLSD), which can be efficiently implemented via the Viterbi algorithm (VA) [41; 162].\(^3\) If the channel \( \{h_l\}_{l=0}^{L} \) is unknown to the receiver, then an estimate of it, \( \{\hat{h}_l\}_{l=0}^{L} \), is usually first formed by using received pilot data, resulting in an adaptive equalization scheme. Such adaptive equalization is implemented in a variety of contemporary systems (the global system for mobile communications (GSM), for example [100]).

Adaptive equalization relies on the estimation of \( \{h_l\}_{l=0}^{L} \), and it therefore requires \( L \) to be known. However, \( L \) may be unknown in practice, and also it may vary with the propagation environment. For example, the equalizer in a GSM phone typically assumes that \( L = 4 \), allowing for delay spreads of up to 20 µs, which may arise in hilly terrain environments [36]. However, in other environments (urban, for instance) one usually encounters much smaller delay spreads and consequently \( L < 4 \). In such cases, an adaptive equalizer based on the assumption that \( L = 4 \) is suboptimal because it tries to estimate a larger number of parameters than what is necessary to uniquely parameterize the channel impulse response.\(^4\) In propagation environments with lower delay spreads, using a smaller \( L \) (i.e., assuming a shorter impulse response) would result in a smaller number of parameters to estimate, which in turn would reduce the total channel estimation error and hence improve the equalizer performance.

\(^1\)The purpose of transmitting the known preambles and postambles typically is to facilitate equalization, and in some cases to provide the necessary means for extracting multipath diversity. In some systems, the data blocks are sequentially transmitted so that the postamble of one block overlaps with the preamble of the next.

\(^2\)In practice, \( L \) is not finite because the channel typically is bandlimited. However, it is common practice to truncate the channel at its effective length.

\(^3\)Unless otherwise is clear from the context, whenever we simply say “equalization” or “adaptive equalization” we refer to (adaptive) equalization using MLSD.

\(^4\)Using the terminology in the system identification literature [140], we would say that the model used by an adaptive equalizer assuming \( L = 4 \) is not parsimonious.
6.1. Introduction

6.1.2 Previous and Related Work

The possible benefit of adapting the number of taps in the equalizer to the propagation environment has been recognized before. For example, [77] suggested to switch between a receiver with an equalizer and a receiver without an equalizer, based on the estimated channel impulse response. Even such a somewhat ad hoc modification can provide a non-negligible improvement of the receiver sensitivity. Another approach, suggested in [1], is to find the number of taps by applying F-tests on a full channel model estimated using frequency domain approaches. Yet, to our knowledge, the problem of adaptive equalization with unknown channel lengths has not been systematically studied. We should note, however, that the related problem of adaptive filtering and prediction with unknown model orders (which is fundamentally different in nature as it neglects the finite-alphabet properties of a communications signal) has been studied for several decades [21; 93; 137] (see also Chapters 3 and 5 herein).

One could conceive a receiver that would first try to determine an estimate of $L$, say $\hat{L}$, and then estimate a channel impulse response of length $\hat{L} + 1$. The estimation of $L$ would then essentially be a model order selection problem. Several order selection methods exist; e.g., see the books [19; 23; 90; 122; 140] and the review papers [82; 150]. Some methods, for example the Akaike information criterion (AIC) [4], are based on minimization of the Kullback-Leibler distance [80] between the observed data and the model. Another approach is to use information-theoretic arguments to determine the length of the shortest binary sequence that can describe the data, leading to the minimum description length (MDL) criterion [119]. A third avenue is to use a Bayesian approach, leading to the Bayesian information criterion (BIC) [73; 127]. BIC is sometimes argued to be preferable as it yields consistent order estimates: One can show that under certain assumptions the probability of selecting the true model approaches 1 as the number of samples $N \rightarrow \infty$. However, even if an “optimal” estimate $\hat{L}$ of the model order $L$ were available, using the model associated with $\hat{L}$ for inference on the data symbols would be suboptimal because information is lost when one assumes that $L = \hat{L}$. The inferences which could have been obtained when assuming other values of $L$ are neglected [64].

6.1.3 Contributions

We present a systematic approach to the problem of equalization for channels with impulse responses of unknown length. The approach circumvents the explicit estimation of $L$, and therefore avoids the fundamental problems associated with methods based on model order selection combined with parameter (channel) estimation (as described above). The key idea of our approach is to parameterize the propagation channel via a Gaussian mixture model.\footnote{This consistency property is not shared by the AIC, but in practice this is of little concern as the models being used seldom perfectly describe the highly complex reality. Furthermore, large data samples are often needed for the consistency property to become visible [19].}

\footnote{It is noteworthy that estimation and detection methods based on mixture models are quite common in the statistical literature [64; 163], and in the literature on pattern recognition and artificial intelligence [170]. However, such models appear to have been much more common in the field of communications than in other fields.}

Based
on the Gaussian mixture model, we derive the maximum a posteriori (MAP) detector, which is known to minimize the probability of a detection error. The mixture model contains a number of hyperparameters (namely, probabilities that the channel is of length \( l = 0, \ldots, L \), and the associated propagation coefficients), which we estimate from a received training sequence (similarly to how the propagation channel is estimated from pilots in a conventional adaptive equalizer). We also present computationally efficient approximations to our detector. Our approach only affects the receiver design, and it is therefore fully compatible with existing transmission schemes.

Notation: Throughout the chapter, \( \Pi_X = X(X^H X)^{-1} X^H \) is the orthogonal projector onto the column space of \( X \); and \( \Pi_{\perp X} = I - \Pi_X \) is the corresponding projector onto the orthogonal complement of the column space of \( X \).

6.2 Preliminaries

To derive the optimal detector it is convenient to first establish a matrix-algebraic framework for channels with intersymbol interference. Towards this end, note that (6.1) can be written as

\[
y = Sh + e
\]  

where

\[
y = \begin{bmatrix} y(0) & \cdots & y(N + L - 1) \end{bmatrix}^T \in \mathbb{C}^{N+L}
\]

\[
e = \begin{bmatrix} e(0) & \cdots & e(N + L - 1) \end{bmatrix}^T \in \mathbb{C}^{N+L}
\]

\[
h = \begin{bmatrix} h_0 & \cdots & h_L \end{bmatrix}^T \in \mathbb{C}^{L+1}
\]

and \( S \) is the following Toeplitz matrix:

\[
S = \begin{bmatrix}
s(0) & s(-1) & \cdots & s(-L) \\
s(1) & s(0) & \cdots & s(1 - L) \\
\vdots & \vdots & \ddots & \vdots \\
s(L) & s(L - 1) & \cdots & s(0) \\
\vdots & \vdots & \ddots & \vdots \\
s(N + L - 1) & s(N + L - 2) & \cdots & s(N - 1)
\end{bmatrix} \in \mathbb{C}^{(N+L)\times(L+1)}. \tag{6.4}
\]

We assume throughout that the noise \( e(n) \) is white, zero-mean circularly symmetric Gaussian with variance \( \sigma^2 \); then \( e \sim \mathcal{CN}(0, \sigma^2 I) \). We also assume that \( s \) and \( e \) are independent, where we introduce the notation \( s = [s(0) \cdots s(N - 1)]^T \). Finally, we let \( S \) be the set of all possible vectors \( s \) (note again that all symbols are chosen from a finite alphabet).

The detector that minimizes the probability of an incorrectly detected symbol sequence \( s \) is the maximum a posteriori detector, i.e., the detector that maximizes \( P(s|y) \) \cite{169}. If all symbol sequences \( \{s\} \) are equiprobable, then

less frequently used for applications in communication theory. See, e.g., \cite{118} for a general discussion of mixture models, in particular Gaussian mixtures.
\[ s_{\text{MAP}} = \arg\max_{s \in S} P(s|y) = \arg\max_{s \in S} p(y|s), \] where \( p(y|s) \) is the likelihood function of \( s \). If \( \hat{h} \) were known to the receiver, then under the assumptions made, optimal detection of \( s \) would amount to minimizing the Euclidean distance

\[ \| y - \hat{S}h \|^2 = \sum_{n=0}^{N+L-1} y(n) - \sum_{l=0}^{L} h_l s(n - l) \]

which can be accomplished by using the Viterbi algorithm. Inserting a suitable estimate \( \hat{h} \) of \( h \) (obtained, for example, via maximum-likelihood) in (6.5) yields a conventional adaptive MLSD equalizer. While such a receiver is in no way optimal, not even when \( \hat{h} \) is an “optimal” estimate of \( h \), it usually works in a satisfactory manner and it has become common practice (see, e.g., Chapter 11 of [114] for a general discussion of adaptive equalizers).

### 6.3 Gaussian Mixture Channel Model

The assumption that underlies our new approach is that the propagation channel \( h \) can be parameterized via a Gaussian mixture model that has one component \( \mathcal{M}_n \) (with the associated probability \( P(\mathcal{M}_n) \)) for each possible channel length \( n = 0, \ldots, L \):

\[
\begin{align*}
\mathcal{M}_0 : & \quad h_0 \sim \mathcal{CN}(\mu_0, \Lambda_0); \quad h_1 = \cdots = h_L = 0 \\
\mathcal{M}_1 : & \quad [h_0 \ h_1]^T \sim \mathcal{CN}(\mu_1, \Lambda_1); \quad h_2 = \cdots = h_L = 0 \\
& \vdots \\
\mathcal{M}_L : & \quad [h_0 \ \cdots \ h_L]^T \sim \mathcal{CN}(\mu_L, \Lambda_L). \tag{6.6}
\end{align*}
\]

(Hereafter we let \( L \) denote the maximum possible channel length; hence, there are \( L + 1 \) possible models.) To interpret the mixture model, one can think that the mechanism which generates \( h \) first chooses the channel length \( n \) (with probability \( P(\mathcal{M}_n) \)) and then generates a random channel \( h \) that has the statistical distribution associated with \( \mathcal{M}_n \). Specifically, with probability \( P(\mathcal{M}_0) \), the channel is frequency-flat; with probability \( P(\mathcal{M}_1) \), the channel has two taps; and with probability \( P(\mathcal{M}_n) \) it has \( n + 1 \) taps. Note that the total pdf of \( h \) is a Gaussian mixture density. Of course, if one knows \textit{a priori} that the channel length is contained in a given interval, one can restrict the number of possible models to a subset of those in (6.6). In this chapter we will assume for simplicity that all models in (6.6) are permissible; the extension to the general case is immediate but it does require some additional notation.

We wish to stress that although at this point it appears that our model would be valid only for channels that actually are Gaussian (or mixture Gaussian), we later (see Section 6.4.1) show that if no \textit{a priori} information on the true channel is available at all, then conditioned on a received training sequence the true channel will have a Gaussian density. Therefore, the proposed mixture Gaussian model is extremely general and it does \textit{not} require the true channel to have a mixture Gaussian density \textit{a priori}. 
In practice, \( \{ \mu_n, \Lambda_n, P(M_n) \} \) are unknown. However, as a preparation, in the next section we first assume that we know these quantities and derive the optimal detector under this assumption. Later, in Section 6.4, we present a method for estimating \( \{ \mu_n, \Lambda_n, P(M_n), \sigma^2 \} \) from a received training (pilot) sequence, thereby obtaining what we call a “trained multi-model detector.”

6.3.1 The Optimal Receiver for Known \( \{ \mu_n, \Lambda_n, P(M_n), \sigma^2 \} \)

We next derive the maximum a posteriori (MAP) detector for the mixture model, assuming that \( \{ \mu_n, \Lambda_n, P(M_n), \sigma^2 \} \) are known and that all symbol sequences \( s \) are equally probable a priori (see the previous section). We have that

\[
p(y|s) = \sum_{n=0}^{L} P(M_n)p(y|s, M_n) \]

\[
= \sum_{n=0}^{L} P(M_n) \int_{\mathbb{C}^n} p(y|h_n, s) p(h_n) dh_n \]

\[
= \sum_{n=0}^{L} P(M_n) \int_{\mathbb{C}^n} \frac{1}{\pi \sigma^2} \frac{1}{\pi^m + 1} d\Lambda_n \]

\[
\cdot \exp \left( -\frac{1}{\sigma^2} \| y - S_n h_n \|^2 - (h_n - \mu_n)^H \Lambda_n^{-1} (h_n - \mu_n) \right) dh_n
\]

where \( S_n \) is the matrix made from the first \( n + 1 \) columns of the matrix \( S \) in (6.4), and \( h_n = [h_0 \cdots h_n]^T \). The integral in (6.7) can be evaluated by some straightforward but somewhat tedious algebra. Note first that

\[
\| y - S_n h_n \|^2 = (h_n - \bar{h}_n)^H S_n^H S_n (h_n - \bar{h}_n) + y^H \bar{\Sigma}_n y
\]

where \( \bar{h}_n = (S_n^H S_n)^{-1} S_n^H y \). It follows that

\[
\| y - S_n h_n \|^2 + \sigma^2 (h_n - \mu_n)^H \Lambda_n^{-1} (h_n - \mu_n)
\]

\[
= (h_n - \bar{h}_n)^H (S_n^H S_n) (h_n - \bar{h}_n) + \sigma^2 (h_n - \mu_n)^H \Lambda_n^{-1} (h_n - \mu_n) + y^H \bar{\Sigma}_n y
\]

\[
= (h_n - \bar{h}_n)^H (\sigma^2 \Lambda_n^{-1} + S_n^H S_n) (h_n - \bar{h}_n) + \Delta + y^H \bar{\Sigma}_n y
\]

where we defined

\[
h_n = (\sigma^2 \Lambda_n^{-1} + S_n^H S_n)^{-1} (\sigma^2 \Lambda_n^{-1} \mu_n + S_n^H S_n \bar{h}_n)
\]

and

\[
\Delta = \bar{h}_n^H S_n^H S_n \bar{h}_n + \sigma^2 \mu_n^H \Lambda_n^{-1} \mu_n
\]

\[
- (\sigma^2 \Lambda_n^{-1} \mu_n + S_n^H S_n \bar{h}_n)^H (\sigma^2 \Lambda_n^{-1} + S_n^H S_n)^{-1} (\sigma^2 \Lambda_n^{-1} \mu_n + S_n^H S_n \bar{h}_n)
\]

\[
= y^H \bar{\Sigma}_n y + \sigma^2 \mu_n^H \Lambda_n^{-1} \mu_n
\]

\[
- (\sigma^2 \Lambda_n^{-1} \mu_n + S_n^H y)^H (\sigma^2 \Lambda_n^{-1} + S_n^H S_n)^{-1} (\sigma^2 \Lambda_n^{-1} \mu_n + S_n^H y).
\]
6.4. Training-Based Estimation of the Mixture Model Parameters

Consequently,

\[
p(y|s) = \frac{1}{(\pi \sigma^2)^{N+L}} \sum_{n=0}^{L} P(M_n) \frac{\sigma^{2(n+1)|\Lambda_n^{-1}|}}{\sigma^2 \Lambda_n^{-1} + S_n^H S_n} \left( -\frac{\Delta + y^H s_n}{\sigma^2} \right) \int_{C_n} \frac{1}{(\pi \sigma^2)^{n+1}} \frac{\sigma^{2n+1}|\Lambda_n^{-1}|}{\sigma^2 \Lambda_n^{-1} + S_n^H S_n} e^{-\frac{1}{2} (h_n - \mu_n)^H \sigma_n^{-1} (h_n - \mu_n)} d\mu_n
\]

\[= \frac{1}{(\pi \sigma^2)^{N+L}} \sum_{n=0}^{L} P(M_n) \frac{\sigma^{2(n+1)|\Lambda_n^{-1}|}}{\sigma^2 \Lambda_n^{-1} + S_n^H S_n} \left( -\frac{\Delta + y^H s_n}{\sigma^2} \right) \exp \left( -\frac{\|y\|^2 + \sigma^2 (\mu_n - h_n)^H (\sigma^2 \Lambda_n^{-1} + S_n^H S_n)^{-1} (\sigma^2 \Lambda_n^{-1} + S_n^H y)}{\sigma^2} \right) \quad (6.12)
\]

(The integrand in (6.12) is the probability density function of a Gaussian random variable, which integrates to unity over \(C_n\).) In conclusion, we have that

\[
\hat{s}_{\text{MAP}} = \arg\max_{s \in S} \left\{ \sum_{n=0}^{L} P(M_n) \frac{\sigma^{2(n+1)|\Lambda_n^{-1}|}}{\sigma^2 \Lambda_n^{-1} + S_n^H S_n} \left( -\frac{\Delta + y^H s_n}{\sigma^2} \right) \exp \left( -\frac{\|y\|^2 + \sigma^2 (\mu_n - h_n)^H (\sigma^2 \Lambda_n^{-1} + S_n^H S_n)^{-1} (\sigma^2 \Lambda_n^{-1} + S_n^H y)}{\sigma^2} \right) \right\} \quad (6.13)
\]

6.3.2 Discussion of the Optimal Receiver

The detection rule (6.13) is optimal (under the assumed model) in the sense that it minimizes the probability that the detected sequence differs from the transmitted one. However, in general one does not know the \textit{a priori} probabilities \(\{P(M_n)\}\), nor the quantities \(\{\mu_n, \Lambda_n, \sigma^2\}\). Since (6.13) is a Bayesian detector, one could think of treating \(\{P(M_n), \mu_n, \Lambda_n, \sigma^2\}\) as hyperparameters and concentrating them out by maximizing (6.12) with respect to \(\{P(M_n), \mu_n, \Lambda_n, \sigma^2\}\). Although this makes sense intuitively we do not pursue this approach here, since maximizing (6.12) with respect to \(\{P(M_n), \mu_n, \Lambda_n, \sigma^2\}\) (and \(s\)) appears to be an intractable problem. Our trained multi-model detector, presented next, provides a way of estimating \(\{P(M_n), \mu_n, \Lambda_n, \sigma^2\}\) from a received training sequence, thereby obviating the problems associated with the direct use of (6.13).

6.4 Training-Based Estimation of the Mixture Model Parameters

We next describe how the parameters (viz., \(\{P(M_n), \mu_n, \Lambda_n, \sigma^2\}\)) in the multi-model detector (6.13) can be estimated from a training sequence. Specifically, we will show that if the noise received during the training sequence is Gaussian,
and a linear estimator is used, then the estimates \( \{ \hat{h}_n \} \) of \( \{ h_n \} \) are Gaussian too and the covariance matrices \( \{ \Lambda_n \} \) are easily obtained. If \( h_n \) is Gaussian conditioned on \( h_n \), then we show that \( h_n \) is Gaussian conditioned on the training data, and hence the Gaussian mixture model assumption made in (6.6) is satisfied. Finally, we use a theorem in [127] to estimate \( P(M_n) \).

6.4.1 Estimation of \( \{ \mu_n, \Lambda_n, \sigma^2 \} \) from a Training Sequence

Suppose we transmit a training sequence \( s_t(n) \) and use it to estimate the channel at the receiver. The received data during the training sequence, of length \( N_t \), can be written

\[
y_t = S_t h + e_t
\]

where \( y_t \) and \( e_t \) are defined in an obvious way and where \( S_t \) is the matrix in (6.4) associated with \( s_t(n) \). For a given model \( M_n \), and assuming Gaussian noise, it is well known that under the assumptions made above, the ML estimate of \( h_n \) is given by

\[
\hat{h}_n = (S_{tn}^H S_{tn})^{-1} S_{tn}^H y_t
\]

where \( S_{tn} \) is the matrix made from the first \( n \) columns of \( S_t \). Since \( \hat{h}_n = h_n + (S_{tn}^H S_{tn})^{-1} S_{tn}^H e_t \), it directly follows that

\[
\hat{h}_n \sim \mathcal{CN} \left( h_n, \sigma^2 (S_{tn}^H S_{tn})^{-1} \right)
\]

conditioned on \( h_n \).

To use the mixture model detector, we need to find the density of \( h \) conditioned on \( \hat{h} \) (i.e., we need to “invert” the conditional density \( p(h|h) \) in (6.16)). Towards this end, let us assume that \( h \) is a random variable of which we have no prior knowledge (before observing the training). Let us model the a priori density of this random variable via \( h \sim \mathcal{CN}(0, \gamma I) \), where \( \gamma \) is a very large positive number. Under this assumption, \( h \) and \( \hat{h} \) are jointly Gaussian with zero mean and covariance matrix

\[
E \left\{ \begin{bmatrix} h \\ \hat{h} \\ h^H \hat{h} \end{bmatrix} \right\} = \begin{bmatrix} \gamma I & \gamma I & \gamma I \\ \gamma I & \gamma I + \sigma^2 (S_{tn}^H S_{tn})^{-1} \end{bmatrix}.
\]

(6.17)

It follows that conditioned on \( \hat{h} \), \( h \) is Gaussian with mean \( \gamma I (\gamma I + \sigma^2 (S_{tn}^H S_{tn})^{-1})^{-1} h \) and covariance matrix \( \gamma I - \gamma I (\gamma I + \sigma^2 (S_{tn}^H S_{tn})^{-1})^{-1} \gamma I \). In the limit when \( \gamma \to \infty \) (corresponding to the case of no a priori information on \( h \)), we find that

\[
\lim_{\gamma \to \infty} \text{cov}(h|\hat{h}) = \lim_{\gamma \to \infty} \left[ \gamma I - \gamma I (\gamma I + \sigma^2 (S_{tn}^H S_{tn})^{-1})^{-1} \gamma I \right] = \sigma^2 (S_{tn}^H S_{tn})^{-1}.
\]

(6.18)

This means that asymptotically as \( \gamma \to \infty \), the density of \( h \) conditioned on \( \hat{h} \) is formally equal to that of \( h \) conditioned on \( h \):

\[
h_n \sim \mathcal{CN} \left( \hat{h}_n, \sigma^2 (S_{tn}^H S_{tn})^{-1} \right)
\]

conditioned on \( y_t \) and \( S_{tn} \).
This result means that the true channel is Gaussian conditioned on the received training, and therefore that the assumptions of the mixture model are satisfied.

From (6.19) we get the values of $\mu_n$ and $\Lambda_n$ to use in (6.13):

$$
\begin{align*}
\mu_n &:= \hat{h}_n = (S^H_{tn}S_{tn})^{-1}S^H_{tn}y_t \\
\Lambda_n &:= \hat{\sigma}^2_n(S^H_{tn}S_{tn})^{-1} 
\end{align*}
$$

where we get the estimated noise variance from the residual:

$$
\hat{\sigma}^2_n = \frac{1}{N_t} ||y_t - S_{tn}\hat{h}_n||^2 = \frac{1}{N_t} ||\Pi_{\perp S_{tn}}y_t||^2. \tag{6.20}
$$

### 6.4.2 Estimation of \( \{P(M_n)\} \)

To estimate the probabilities \( \{P(M_n)\} \), we use a result from [127] (also presented in a more popularized form in Chapters 2 and 5). Under some assumptions, one can show that

$$
\ln p(y_t|M_n) \approx -\frac{1}{2}BIC_n + \text{const.} \tag{6.21}
$$

where

$$
BIC_n \triangleq -2\ln[p(y_t|M_n)]_{h_n:=\hat{h}_n} + (2(n + 1) + 1) \ln 2N_t \tag{6.22}
$$

and where the approximation error in (6.21) goes to zero when the amount of training increases without bound. In (6.22), \([p(y_t|M_n)]_{h_n:=\hat{h}_n}\) is the likelihood function with $h_n$ replaced by $\hat{h}_n$, and $\sigma^2$ replaced by $\hat{\sigma}^2$ (6.20); that is (cf. (2.8)),

$$
[p(y_t|M_n)]_{h_n:=\hat{h}_n} = \text{const.} \cdot \frac{1}{(\hat{\sigma}^2_n)^{N_t}}. \tag{6.23}
$$

Hence, to within an additive constant,

$$
BIC_n = 2N_t \ln ||\Pi_{\perp S_{tn}}y_t||^2 + (2(n + 1) + 1) \ln 2N_t. \tag{6.24}
$$

It is clear that

$$
P(M_n|y_t) = \frac{p(y_t|M_n)P_t(M_n)}{p(y_t)} \tag{6.25}
$$

where $P_t(M_n)$ stands for the probability of $M_n$ \textit{a priori} with respect to the training data. Assuming that all models (channel lengths) are equally probable \textit{a priori} (before observing the training), i.e., $P_t(M_n) = 1/(L+1)$, and using the fact that $\sum_{n=0}^L P(M_n|y_t) = 1$, (6.25) and (6.21) lead to the following estimate of the \textit{a posteriori} (with respect to the training data) probability:

$$
\hat{P}(M_n|y_t) = \frac{\exp\left(-\frac{1}{2}BIC_n\right)}{\sum_{k=0}^L \exp\left(-\frac{1}{2}BIC_k\right)} \tag{6.26}
$$

where $BIC_k$ is given by (6.24). The above estimate is \textit{a priori} with respect to the data vector $y$ and can thus be used as an estimate of $P(M_n)$ in (6.13).
6. The Trained Multi-Model Detector and Its Implementation

In this section we first discuss the detector that results from using estimates of \( \{\mu_n, \Lambda_n, P(M_n), \sigma^2\} \) in (6.13). We then go on to discuss approximations of the so-obtained adaptive detector.

6.5.1 Trained Multi-Model Detector

We have found in Section 6.4.1 that, conditioned on the observed training data \( y_t \), the channels \( \{h_n\} \) are Gaussian distributed according to (6.19). Thus, the assumptions of the Gaussian mixture model (6.6) are satisfied. Furthermore, given \( \{y_t, S_t\} \), (6.19), (6.20) and (6.26) provide estimates of \( \{\mu_n, \Lambda_n, P(M_n), \sigma^2\} \). This means that we can apply the multi-model detector (6.13) with

\[
\mu_n := \hat{h}_n = \left( S_{tn}^H S_{tn} \right)^{-1} S_{tn}^H y_t \\
\Lambda_n := \hat{\sigma}^2 \left( S_{tn}^H S_{tn} \right)^{-1} \\
\sigma^2 := \hat{\sigma}^2_L = \frac{1}{N_t} \| \Pi_{S_{tn}} y_t \|^2 \\
P(M_n) := \frac{\exp \left( -\frac{1}{2} BIC_n \right)}{\sum_{k=0}^{L} \exp \left( -\frac{1}{2} BIC_k \right)}
\]

(6.27)

where \( BIC_k \) is given by (6.24). Note that the estimates in (6.27) are a posteriori with respect to the training data but a priori with respect to the unknown symbol data sample.

In (6.27), we use the noise variance estimate associated with the maximum channel order, viz. \( \hat{\sigma}^2_L \). We choose to do so because \( \hat{\sigma}^2_L \) is usually a “safer” estimate of \( \sigma^2 \) than \( \hat{\sigma}^2_n \) is. Indeed, when \( n \) is less than the true channel length, then the more “natural” estimate \( \hat{\sigma}^2_n \) tends to be an overestimate of \( \sigma^2 \). This is so because in this case, \( \hat{\sigma}^2_n \) will be the sum of the noise variance and a component associated with the residual modeling error caused by the use of an insufficient number of parameters.

The detector resulting from (6.13) and (6.27) can be explicitly spelled out (by inserting (6.27) into (6.13)), but in the interest of brevity we avoid presenting the resulting, somewhat messy expression. We stress that the so-obtained detector is not fully optimal, despite the optimality of (6.13) for known \( \{\mu_n, \Lambda_n, P(M_n), \sigma^2\} \); indeed, the procedure of first extracting information from the training and then using that information in the detector (6.13) is suboptimal in itself. Nevertheless, it is noteworthy that the detector (6.13)+(6.27) extracts more information (viz., \( \{\mu_n\}, \{\Lambda_n\}, \sigma^2 \) and \( \{P(M_n)\} \)) from the training than what a conventional training-based detector typically does; indeed, the latter typically only determines a fixed-order channel estimate \( \hat{h} \) and uses it in the detector as if it were equal to the true channel. Such a conventional training-based detector is a special case that we can obtain by successively simplifying the equations (6.13) and (6.27), as explained in Section 6.5.2.
As a further remark, note that when we derived the optimal MAP detector in (6.13), we assumed that $\sigma^2$ was a deterministic constant. However, the estimate of $\sigma^2$, viz. $\hat{\sigma}^2$, obtained from the training is a random variable. One could therefore think of obtaining the distribution of $\sigma^2$ conditioned on the training data, and appropriately modify the detector (6.13) to optimally make use of this information as well. However, we do not pursue this path. First, since the noise variance can usually be fairly accurately estimated, and since symbol detection criteria do not seem to be excessively sensitive to the estimation of noise powers in general, we expect the gain obtained by incorporating the distribution of $\sigma^2$ (conditioned on the training data) into (6.13) to be small. Second, it appears to be difficult to obtain the associated MAP detector in closed form.

### 6.5.2 Possible Approximations of the Trained Multi-Model Detector (6.13)+(6.27)

The problem of directly solving (6.13), using (6.27), is computationally rather hard. One can obtain simplified forms of this problem by performing a series of approximations as described next.

**Approximation 1 ($h_n = \hat{h}_n$):** The first possible approximation is to assume that $h_n$ is perfectly estimated from the training. In loose words, this amounts to assuming that $\mu_n = \hat{h}_n$ and $\Lambda_n = 0$, whereby the distribution of $\hat{h}_n$ becomes a Dirac-impulse at $h_n = \hat{h}_n$. In this case, the integral over $h_n$ disappears from (6.7), and we are left with:

$$p(y|s) = \sum_{n=0}^{L} P(M_n) p(y|\hat{h}_n, s) \frac{1}{(\pi\sigma^2)^{N+L}} \sum_{n=0}^{L} P(M_n) \exp \left( -\frac{1}{\sigma^2} \| y - S_n \hat{h}_n \|^2 \right).$$

(6.28)

**Approximation 2 (Weighted LS metric):** Since the computational complexity of (6.28) is prohibitively high for long sequences, one can also think of the following further approximation,

$$\hat{s} = \arg \min_s \left\{ \sum_{n=0}^{L} P(M_n) \| y - S_n \hat{h}_n \|^2 \right\}$$

$$= \arg \min_s \left\{ \sum_{k=0}^{N+L-1} \left( \sum_{n=0}^{L} P(M_n) \left| y(k) - \sum_{l=0}^{n} \hat{h}_n^l s(k-l) \right| \right)^2 \right\}$$

(6.29)

which is easily implemented by a standard VA using a vector-valued branch metric. The approximation involved in going from (6.28) to (6.29) is rather ad hoc, and indeed only motivated by the fact that (6.29) resembles an order-averaged least-squares metric.
Approximation 3 ($M_n = M_{\hat{n}}$): As another further possible approximation of (6.28), one could assume that the model $M_n$ with the highest a posteriori probability $P(M_n)$ given by (6.26) is the true one, thereby discarding most of the information in $\{\hat{P}(M_n)\}$. This would correspond to first estimating the channel length and the associated impulse response, and then using these estimated quantities in lieu of the unknown ones. In this case, the sum over $n$ in (6.7) disappears, and one is left with the decision metric

$$\hat{s} = \arg\min_{\hat{s} \in S} \left\| y - S_{\hat{n}}\hat{h}_n \right\|^2 = \arg\min_{\hat{s}} \left\{ \sum_{k=0}^{N+L-1} \left| y(k) - \sum_{l=0}^{\hat{n}} \hat{h}_n^l s(k-l) \right|^2 \right\}$$

(6.30)

where

$$\hat{n} = \arg\max_{n \in \{0, \ldots, L\}} \hat{P}(M_n | y_t) = \arg\min_{\hat{n}} \text{BIC}_n.$$  

(6.31)

Equation (6.30) is the conventional MLSD associated with the most likely model and the solution can be easily obtained with a standard VA. Soft-decisions could also be obtained by using a soft-output Viterbi algorithm (SOVA) [57] on (6.30). For more details on implementing a SOVA for this problem, see [87].

6.6 Numerical Examples

We illustrate the error performance of the previously introduced detectors via Monte-Carlo simulation. We use a simple channel model to compare the performances of the trained multi-model detector (see Section 6.5.1) and the possible approximations of it (see Section 6.5.2).

Example 1 (performance of trained multiple model detector for short simple channel and short data sequence): We use a short data sequence of length $N = 6$ and a channel with $L + 1$ different exponentially decaying delay profiles, all occurring with the same probability $1/(L + 1)$. For the $l$th delay profile ($l \in \{1, 2, \ldots, L + 1\}$):

$$E\left\{ \left| h_0 \right|^2 \left| h_1 \right|^2 \cdots \left| h_L \right|^2 \right\} = \left[ 1 \ e^{-0.25} \cdots \ e^{-0.25(l-1)} \ 0 \ \cdots \ 0 \right] \left( \sum_{k=0}^{L-1} e^{-0.25k} \right)$$

(6.32)

where the scaling factor in the denominator ensures that all delay profiles have unit energy. In this example we have set $L = 2$; hence the channel has up to $L + 1 = 3$ non-zero taps. A 4 samples long sequence, $s_t = [-1, -1, 1, -1]^T$, was used as training sequence for the estimation of $\{P(M_n), \mu_n, \Lambda_n, \sigma^2\}$ (other pilot symbol sequence lengths give similar results, but the difference between the methods tend to decrease when the length of the pilot symbol sequence increases).
6.7 Concluding Remarks

Figure 6.1 shows the bit-error-rate (BER) for: (a) adaptive equalization with a 1 tap channel estimate; (b) adaptive equalization, 2 taps; (c) adaptive equalization, 3 taps; (d) adaptive equalization, 4 taps; (e) adaptive multi-model equalization using (6.13)+(6.27); and (f) coherent detection (using the true channel). Here, (a)–(d) were obtained using (6.30) with $\hat{n} =$ number of taps$-1$. The improvement obtained by using the multi-model equalizer instead of the conventional adaptive MLSD (with 3 taps) is about 1 dB in this example. The performances of the conventional adaptive MLSDs using 1 or 2 taps are poor (of course), but shown for completeness. Note that the adaptive MLSD using 4 taps always uses a non-parsimonious parameterization of the channel, as it has to estimate at least one unnecessary parameter; this is why it performs worse than the MLSD using 3 taps.

In Figure 6.2, we compare the performance of conventional adaptive equalization using 3 taps, adaptive multi-model equalization (using (6.13) + (6.27)), and the various approximations in (6.28), (6.29), and (6.30). The figure shows how successive approximations of the trained multi-model equalizer degrade the performance.

Example 2 (performance of approximated multiple model detector for a long sequence): We next use a longer sequence ($N = 150$) and a channel with up to $L + 1 = 5$ taps with the exponentially decaying delay profiles in (6.32) (with $L = 4$), to compare the performance of conventional Viterbi equalization with that of the approximate multi-model detectors. A 7 samples long sequence, $s_t = [−1, −1, 1, −1, −1, 1, −1]^T$, was used as training signal. Figure 6.3 shows the BER for: (a) conventional adaptive MLSD with a 5 taps channel estimate; (b) adaptive equalization using the estimated model (6.30); (c) detection using (6.29); and (d) coherent detection. The performances of the adaptive multi-model detector given by (6.13)+(6.27) and of the detector (6.28) are not shown, as these detectors are computationally prohibitive in this example. A Viterbi equalizer was used to minimize all decision metrics. In this example as well, the different approximations of the multi-model equalizer (6.13) outperform the conventional adaptive Viterbi equalizer; the improvement is about 1.5 dB for this example. The difference between the two approximations (6.29) and (6.30) (note that the latter is not a multi-model approach, but a single model approach using the model structure selected by BIC) is not very large in this example, but the multi-model approach of (6.29) has a slight advantage.

6.7 Concluding Remarks

We have devised mixture model-based adaptive equalizers for communication channels that can be modeled as finite-impulse-response filters with unknown length. The performances of the proposed receivers were evaluated numerically by using a simple channel model. The proposed receivers have shown promising performance. If the increased computational complexity can be afforded, they appear to be practically viable and attractive possibilities for improving the
Figure 6.1: Comparison of multi-model equalization and conventional adaptive Viterbi equalization for a short sequence ($N = 6$) and a channel with at most three taps ($L = 2$). The pilot sequence had length 4.

sensitivity performance of equalizers in wireless systems.
Figure 6.2: Comparison of different approximations to the adaptive multimodel equalizer for a short sequence ($N = 6$), and a channel with a maximum of three taps ($L = 2$). The pilot sequence had length 4.
Figure 6.3: Comparison of adaptive multi-model equalization and conventional Viterbi equalization for a long sequence ($N = 150$), and a channel with up to five taps ($L = 4$). The pilot sequence had length 7.
Chapter 7

Empirical Bayes Linear Regression with Unknown Model Order

We study maximum a posteriori probability model order selection for linear regression models, assuming Gaussian distributed noise and coefficient vectors. For the same data model, we also derive the minimum mean-square error coefficient vector estimate. The approaches are denoted BOSS (Bayesian Order Selection Strategy) and BPM (Bayesian Parameter estimation Method), respectively. In their simplest form, both BOSS and BPM require a priori knowledge of the distribution of the coefficients. However, under the assumption that the coefficient variance profile is smooth, we derive “empirical Bayesian” versions of our algorithms which estimate the coefficient variance profile from the observations and thus require little or no information from the user. We show in numerical examples that the estimators can outperform several classical methods, including the well-known AICc and BIC for model order selection.

7.1 Introduction

7.1.1 Problem Formulation

Consider the linear regression model

\[ y = Xh + \epsilon \]  

(7.1)

where \( y \in \mathbb{R}^N \) is the vector of observed data, \( X = [x_1 \cdots x_n] \in \mathbb{R}^{N \times n} \) is a known matrix of \( n \) regressors \( \{x_j\}_{j=1}^n \), \( h = [h_1 \cdots h_n]^T \in \mathbb{R}^n \) is the unknown
vector of linear regression coefficients (we will also sometimes call \( h \) the parameter vector) and \( \epsilon \sim \mathcal{N}(0, \sigma^2 I) \) is a length \( N \) vector of zero-mean Gaussian white noise with variance \( \sigma^2 \). We call (7.1) the full model and assume that the data are generated by a model of the form

\[
\mathcal{M}_k : y = X_k h_k + \epsilon
\]

(7.2)

where \( n_{\text{min}} \leq k \leq n \), \( X_k = [x_1 \cdots x_k] \) (i.e., \( X_k \) consists of the first \( k \) columns of \( X \)), and \( h_k = [h_1 \cdots h_k]^T \). Furthermore, we make the assumption that the coefficients \( h_j \) are zero-mean independent Gaussian random variables, \( h_j \sim \mathcal{N}(0, \gamma_j^2) \). In other words, \( h_k \sim \mathcal{N}(0, \Gamma_k) \) where \( \Gamma_k = \text{diag}[\gamma_1^2 \cdots \gamma_k^2] \). We assume that the model order \( k \) and the variances \( \{\gamma_j^2\}_{j=1}^k, \sigma^2 \) are unknown.

We consider the following two classical interrelated problems:

1. The model order selection problem: to find the correct order \( k \), given \( X \) and \( y \).
2. The parameter estimation problem: to estimate \( h \) as accurately as possible when the order \( k \) is unknown.

### 7.1.2 Related Work

Bayesian solutions to the above two problems, under the Gaussianity assumption on the coefficients and the noise, are available in the literature. In, e.g., [55], the maximum likelihood (ML) and maximum a posteriori (MAP) model order selection algorithms for the current model were derived, although not numerically evaluated. In [63], the minimum mean square error (MMSE) estimate of a frequency function was derived. Within the same framework, it is easy to derive the MMSE estimate of \( h \).

Classical approaches to model selection include the information criteria, which will be used for comparative purposes in our numerical examples in Section 7.4. However, they are not based on the Bayesian paradigm and are thus less related to the current approach than the methods discussed above. For a review of the information criterion approach, see Chapter 2.

The problem of determining the number of signal components in a mixture is fundamentally important also for many other types of data models. For example in array processing for determining the number of sources impinging on an array of sensors [101], or in line spectrum analysis for determining the number of sinusoids in noise [30; 44]. In the present chapter, however, we are only concerned with the linear regression data model (7.1).

### 7.1.3 Contributions of This Work

For the Bayesian approaches in the above subsection it is generally assumed that appropriate values for the noise variance \( \sigma^2 \) and the coefficient variances \( \{\gamma_j^2\}_{j=1}^n \) are known. This assumption does not necessarily hold in applications.

---

1The empirical Bayesian estimators to be presented in Section 7.3 require \( n < N \), but the theory presented in Sections 7.1-7.2 does not need this assumption.
The goal of the present chapter is to present methods which do not require knowledge of \( \sigma^2 \) and \( \{\gamma_j^2\}_{j=1}^n \). To this end we take an empirical Bayes approach: we estimate \( \sigma^2 \), \( \{\gamma_j^2\}_{j=1}^n \) from the data and then use the resulting estimates as if they were the true values. See Section 7.3.

Note that the models in (7.2) that we consider are nested, in the sense that a lower order model can be obtained as a special case of a higher order model (by setting certain coefficients to zero). This type of models is common in signal processing applications, such as finite-impulse-response (FIR) filter identification or the estimation of polynomial coefficients. By way of contrast, it is common in statistical data analysis to consider sparse models for which the true model can consist of any subset of the regressors \( \{x_k\}_{k=1}^n \). We consider linear regression for sparse models in Chapter 9 where we, among other things, propose an empirical Bayesian technique tailored to sparse models. However, for the non-sparse unknown model order problem, the technique in Chapter 9 is less sophisticated and flexible than the one we provide in this chapter.

## 7.2 Bayesian Model Order Selection and Parameter Estimation

### 7.2.1 Optimal Model Order Selection

Here we review the MAP model order selection algorithm for the problem posed in Section 7.1, assuming known \( \sigma^2 \), \( \{\gamma_j^2\}_{j=1}^n \). This model selection rule has been derived previously in [55]. In the remainder of the chapter this specific model selection algorithm will be denoted by BOSS (Bayesian Order Selection Strategy).

Using Bayes’ Theorem we obtain an expression for the model posterior probabilities:

\[
P(M_k|y) = \frac{P(M_k)p(y|M_k)}{p(y)}.
\]

Since \( p(y) \) is independent of the model \( M_k \), the model order which gives the highest posterior probability is

\[
\hat{k}_{\text{MAP}} = \arg \max_{k = n_{\text{min}}, \ldots, n} P(M_k)p(y|M_k).
\]

If nothing is known about the model prior probabilities \( P(M_k) \), we will assume that they are equal (this is common practice [163]) and, of course, that they sum up to one:

\[
P(M_k) = \frac{1}{n - n_{\text{min}} + 1}, \quad k = n_{\text{min}}, \ldots, n.
\]

Furthermore, under the assumption that \( M_k \) is the data generating model we have

\[
y|M_k \sim \mathcal{N}(0, Q_k)
\]

where

\[
Q_k = X_k \Gamma_k X_k^T + \sigma^2 I, \quad \Gamma_k = \text{diag}[\gamma_1^2, \ldots, \gamma_k^2].
\]
7. Empirical Bayes Linear Regression with Unknown Model Order

So,

\[ p(y|M_k) = \frac{1}{\sqrt{2\pi}} \frac{1}{|Q_k|^{1/2}} \exp \left( -\frac{1}{2} y^T Q_k^{-1} y \right). \tag{7.5} \]

We obtain BOSS by using (7.4) and (7.5) to compute (7.3). One can show (see, e.g., the discussion around (2.36)) that the order with the highest \textit{a posteriori} probability is also the most likely to be the correct order.

Note that (using the identity \(|I + AB| = |I + BA|\))

\[ |Q_k| = |X_k \Gamma_k X_k^T + \sigma^2 I| = \sigma^{2N} \frac{|X_k^T X_k \Gamma_k + I|}{\sigma^2} \tag{7.6} \]

and (using the matrix inversion lemma)

\[ Q_k^{-1} = (X_k \Gamma_k X_k^T + \sigma^2 I)^{-1} = \frac{1}{\sigma^2} I - \frac{1}{\sigma^4} X_k \left( \Gamma_k^{-1} + \frac{1}{\sigma^2} X_k^T X_k \right)^{-1} X_k^T. \tag{7.7} \]

The expressions in (7.6) and (7.7) can be used to boost the computational efficiency: \(X_k^T X_k\) is much faster to evaluate than \(X_k X_k^T\) if \(N \gg n\) (\(N \geq n\) will be a necessary condition for our empirical Bayesian methods to work, see Section 7.3). One should also exploit the fact that \(\Gamma_k\) is diagonal. The computational complexity of (7.6) can be further reduced by noting that

\[ X_k^T X_k = \begin{bmatrix} X_k^T \cr x_k^T \end{bmatrix} \begin{bmatrix} X_{k-1} & x_k \end{bmatrix} = \begin{bmatrix} X_k^T x_{k-1} \cr x_k^T x_{k-1} & x_k^T x_k \end{bmatrix} \]

and that \(X_{k-1}^T X_{k-1}\) is evaluated when computing \(p(y|M_{k-1})\). Also, a well-known result on the inverse of partitioned matrices (see, e.g., Lemma A.2 in [140]) can be used to compute (7.7) iteratively. Define \(Z_k = (\Gamma_k^{-1} + \frac{1}{\sigma^2} X_k^T X_k)\).

Then

\[ Z_k^{-1} = \begin{bmatrix} Z_{k-1}^{-1} & 0_{k-1} \\ 0_{k-1} & 1 \end{bmatrix} + \frac{-Z_{k-1}^{-1} \left( \frac{x_k^T X_{k-1}}{\sigma^2} \right)^T}{\sigma^2} \begin{bmatrix} x_k^T X_k x_k & \frac{x_k^T X_{k-1}}{\sigma^2} Z_{k-1}^{-1} \\ \frac{x_k^T x_k}{\sigma^2} + \frac{1}{\sigma^4} x_k^T X_k x_{k-1} Z_{k-1}^{-1} \left( x_k^T X_{k-1} \right)^T \end{bmatrix}, \tag{7.8} \]

so the matrix inversion in (7.7) needs only be computed once, namely for \(k = n_{\min}\).

7.2.2 The MMSE Estimate of \(h\)

In this section we describe what we will denote the BPM (Bayesian Parameter estimation Method) by deriving the MMSE estimate of \(h\) under the assumption that one of the models \(\{M_k\}_{k=n_{\min}}^n\) in (7.2) generated the data. The MMSE
7.3 Empirical Bayesian Estimation

estimate equals the conditional mean [140]:

$$\hat{h}_{\text{MMSE}} = E[h|y] = \sum_{k=n_{\min}}^{n} P(M_k|y)E[h|y, M_k]$$

$$= \sum_{k=n_{\min}}^{n} P(M_k)p(y|M_k)E[h|y, M_k]$$

$$= \sum_{k=n_{\min}}^{n} P(M_k)p(y|M_k)\frac{\sum_{j=n_{\min}}^{n} P(M_j)p(y|M_j)}{\sum_{j=n_{\min}}^{n} P(M_j)p(y|M_j)}$$  \hspace{1cm} (7.9)

where we used

$$P(M_k|y) = P(M_k)p(y|M_k) = P(M_k)p(y|M_k) \frac{\sum_{j=n_{\min}}^{n} P(M_j)p(y|M_j)}{\sum_{j=n_{\min}}^{n} P(M_j)p(y|M_j)}.$$  

We can use (7.4) and (7.5) in (7.9). What remains to evaluate (7.9) and get the BPM is then to compute $E[h|y, M_k]$.

Clearly, assuming that the model $M_k$ generated the data, $h_j = 0$ for $j > k$, so it is sufficient to find $E[h_k|y, M_k]$. Under $M_k$, $h_k$ and $y$ are jointly Gaussian:

$$\begin{bmatrix} y \\ h_k \end{bmatrix} | M_k \sim \mathcal{N}\left(0, \begin{bmatrix} Q_k & X_k \Gamma_k^T \\ \Gamma_k & \Gamma_k \end{bmatrix} \right).$$

Applying a standard result (Lemma B.17 in [140], for example), the conditional mean evaluates to

$$E[h_k|y, M_k] = \Gamma_k X_k^T Q_k^{-1} y.$$  \hspace{1cm} (7.10)

We now obtain the BPM by inserting (7.4), (7.5) and (7.10) into (7.9).

7.2.3 A Note on Complexity

The computational costs of BOSS and BPM are dominated by the determinant in (7.6) and the matrix inverse (7.7) ($O(k^2 N)$ operations). Both (7.6) and (7.7) have to be performed for $k = n_{\min}, \ldots, n$, so the total complexity of BOSS and BPM is $O(mn^2 N)$ where $m = n - n_{\min} + 1$ is the total number of models considered. For comparison, the classical information criterion approach to model order selection (see the Chapter 2) has, in the scenario considered here, a computational complexity of $O(N(nm + n_{\min}^2))$ (if one makes use of (7.8)).

7.3 Empirical Bayesian Estimation

The weakness of BOSS and BPM is that they require knowledge of $\{\gamma_j^2\}_{j=1}^{n}$ and $\sigma^2$. In practice, suitable values for these variances may be unknown. This problem can be dealt with in different ways.

One possibility is to assign hierarchical distributions to the unknown variances and set the resulting hyperparameters to some values. For example, one may assume that $\sigma^2 \sim \mathcal{D}(a)$ where $\mathcal{D}$ denotes some distribution and $a$ is a hyperparameter. The resulting expressions, corresponding to (7.3) and (7.9),
can typically not be evaluated in closed form. Thus one has to resort to numerical approaches, such as Markov Chain Monte-Carlo (MCMC) methods [92]. For a related example, see [39]. This type of numerical approach is generally computationally intensive. Furthermore, it is not always clear at what hierarchical level one should stop the hyperparameterization or what values should be assigned to the hyperparameters at the final level: In the above example, should \(a\) be set to a specific value (and if so, to what value?), or should we assign yet a distribution for \(a\)?

Instead, we propose to use an empirical Bayes approach. In this approach the unknown prior (hyper)parameters are estimated from the available data. The so-obtained estimates of \(\{\gamma_j^2\}_{j=1}^n, \sigma^2\) can then be used for inference, i.e., be inserted into (7.3) and (7.9) as if they were the true values. This is somewhat against the true spirit of Bayesian inference, as the variances \(\{\gamma_j^2\}_{j=1}^n, \sigma^2\) are \textit{a priori} parameters, which should not depend on the data. Nevertheless, estimation of \(\{\gamma_j^2\}_{j=1}^n, \sigma^2\) appears to be an attractive, pragmatic way of handling the situation when these parameters are completely unknown.

The idea behind the empirical Bayes methods to be presented herein is to replace rather specific \textit{a priori} knowledge about the data (the true values of \(\{\gamma_j^2\}_{j=1}^n, \sigma^2\)) by much less specific \textit{a priori} knowledge. To this end, we will base our estimates of \(\{\gamma_j^2\}_{j=1}^n, \sigma^2\) on the ML/LS (least squares) estimator,

\[
\hat{h}_{\text{ML},k} = \hat{h}_{\text{LS},k} = (X_k^T X_k)^{-1} X_k^T y, \tag{7.11}
\]

of the full model (7.1) (i.e., \(k := n\) in (7.11)). Note that (7.11) requires \(N \geq n\) when \(k = n\) and that \(X\) is full rank. These conditions are necessary for our empirical Bayes methods to work. In fact, the estimate of \(\sigma^2\) that we will use requires \(N > n\) (see below).

The estimation of \(\sigma^2\) is straightforward: an unbiased, consistent estimate of \(\sigma^2\) can be obtained by taking [130]

\[
\hat{\sigma}^2 = \frac{1}{N - n} ||y - X \hat{h}_{\text{LS},n}||^2. \tag{7.12}
\]

The estimation of \(\{\gamma_j^2\}_{j=1}^n\) is more challenging. There are two distinct problems that we need to cope with:

First, only a single data realization is available for estimation of the variances \(\{\gamma_j^2\}_{j=1}^n\). In fact, \(h\) is not directly available but has to be estimated, as in (7.11), from \(y\). To put it differently, only one estimated sample \(\hat{h}_j\) is available for the estimation of the variance \(\gamma_j^2\). This makes the problem of estimating \(\{\gamma_j^2\}_{j=1}^n\) from \(y\) ill-posed.

Second, for some coefficients, there may be no data at all available for the variance estimation. Indeed, if the true order is \(k\), there are no data available for estimation of \(\gamma_j^2\), \(j > k\), since then \(h_j = 0\). If this problem were disregarded we would likely end up with severely underestimated \(\hat{\gamma}_j^2\) for \(j > k\), since the estimates would be based on \(\hat{h}_j \approx 0\). This would give a high risk of overestimating the model order and in turn result in poor estimates of \(h_j, j > k\).
From the above reasoning, it is obvious that we need to introduce some a priori information on the variances \( \{ \gamma_j^2 \}_{j=1}^n \). In [130] it was assumed that \( \gamma_j^2 = \gamma^2 \), \( j = 1, \ldots, n \). Herein, we take a more general approach, as detailed below. We also note that it is likely that any estimates of \( \{ \gamma_j^2 \}_{j=1}^n \) will deviate rather much from the true values. Fortunately, BOSS and BPM appear to be relatively robust against mismatching values of \( \{ \gamma_j^2 \}_{j=1}^n \) (see, e.g., [130] and the numerical examples herein). To cope with the first of the above-mentioned problems we make the (in many applications reasonable) assumption that \( \gamma_j^2 = \gamma^2 \), \( j = 1, \ldots, n \). Herein, we take a more general approach, as detailed below. We also note that it is likely that any estimates of \( \{ \gamma_j^2 \}_{j=1}^n \) will deviate rather much from the true values. Fortunately, BOSS and BPM appear to be relatively robust against mismatching values of \( \{ \gamma_j^2 \}_{j=1}^n \) (see, e.g., [130] and the numerical examples herein).

As a rule of thumb, we recommend \( b_f = 0.1 \). The parameter \( b_f \) can be viewed as a priori knowledge expressing the difference in magnitude between the lowest and greatest variance of the linear regression coefficients. It can also be viewed as a robustification parameter which guarantees that none of the variances \( \{ \gamma_j^2 \}_{j=1}^n \) will be severely underestimated. This suggested approach is clearly ad hoc, but more systematic ways of dealing with the problem are not obvious. Also, we have found via simulations that the choice of \( b_f \) is not critical for performance (see Section 7.4). This is probably related to the relative insensitivity of BOSS/BPM against mismatches in \( \{ \gamma_j^2 \}_{j=1}^n \), as mentioned above.

We will now describe two different ways of incorporating smoothness constraints on \( \gamma_j^2 \).

### 7.3.1 Estimation of \( \{ \gamma_j^2 \}_{j=1}^n \) by Parameterization

In this approach we assume that \( \{ \gamma_j^2 \}_{j=1}^n \) can be expressed as a linear combination of a relatively small number of known basis vectors, as follows:

\[
\gamma = \Psi \alpha, \tag{7.14}
\]

where \( \gamma = [\gamma_1^2 \cdots \gamma_n^2]^T \), \( \Psi \) is a known \( n \times r \) matrix and \( \alpha \) is an unknown vector of length \( r \). The columns of \( \Psi \) should be a basis for “likely” variance profiles for \( \{ \gamma_j^2 \}_{j=1}^n \) and generally \( r \) should be small compared to \( n \). (In a communication channel identification application, e.g., it might be known that \( \{ \gamma_j^2 \}_{j=1}^n \) is exponentially decaying [98], and \( \Psi \) could then contain one or a few columns with different degrees of exponential decay.) If nothing at all is known about \( \{ \gamma_j^2 \}_{j=1}^n \), then one possibility is to construct \( \Psi \) from the first \( r \) basis functions in a discrete Fourier series expansion of \( \gamma \) (in this case, \( r \) is a user parameter):

\[
\begin{align*}
[\Psi]_{i,1} &= 1 \\
[\Psi]_{i,j} &= \begin{cases} 
\cos\left(\frac{j}{2}(i - 1)\frac{2 \pi}{n}\right) & j \text{ even} \\
\sin\left(\frac{j - 1}{2}(i - 1)\frac{2 \pi}{n}\right) & j \text{ odd and } \geq 3
\end{cases}
\end{align*}
\tag{7.15}
\]
where $i = 1, \ldots, n$, $j = 1, \ldots, r$ with $r$ odd, and $[\Psi]_{i,j}$ denotes the element on the $i$th row and $j$th column of $\Psi$. Using a truncated Fourier series as basis may seem ad hoc at first glance, but we find it attractive for many reasons: the basis functions are orthogonal and the resulting variance profile is smooth. Additionally, by choosing the number of basis functions $r$, we can directly influence the amount of variation in the variance profile $\gamma$.

Now, the problem is reduced to that of estimating $\alpha$. Thus, the dimension-
ality of the problem is reduced from $n$ to $r$. Note that

$$E[|h_j|^2] = \sum_{k=n_{\text{min}}}^{n} P(M_k) E[|h_j|^2|\,\mathcal{M}_k] = q_j \gamma_j^2$$  \hspace{1cm} (7.16)

where $q_j = \sum_{k=\max\{j,n_{\text{min}}\}}^{n} P(M_k)$, because under $M_k$, $h_j = 0$ for $j > k$, and $h_j$ has variance $\gamma_j^2$ if $j \leq k$. By replacing $E[|h_j|^2]$ in the above equation by the squared norm of the corresponding element of the full model LS estimate ((7.11) with $k := n$), $|h_{LS,j}|^2$, and using (7.14) we can estimate $\alpha$ from the following constrained LS expression:

$$\hat{\alpha} = \arg \min_{\alpha} \left\| \begin{bmatrix} |\hat{h}_{LS,1}|^2 & \cdots & q_1[\Psi]_{1,1} \\ \vdots & \ddots & \vdots \\ |\hat{h}_{LS,r}|^2 & \cdots & q_r[\Psi]_{1,r} \end{bmatrix} \alpha \right\|^2 \quad \text{s.t.} \quad \Psi \alpha \geq 0.$$  \hspace{1cm} (7.17)

We then get $\hat{\gamma}$ by inserting $\hat{\alpha}$ in (7.14). The constraint in (7.17) guarantees that all $\hat{\gamma}_j^2 \geq 0$. The above expression can be easily and efficiently solved by quadratic programming (use, e.g., \texttt{quadprog} in Matlab).

One might ask why the threshold $b$ is not used already in (7.17) (and (7.18) below). The reason is that the thresholds $\gamma \geq 0$ and $\hat{\gamma}_j^2 \geq b$ serve different purposes: $\gamma \geq 0$ used in (7.17) and (7.18) takes care of the physical requirement that the variances must be non-negative, whereas the threshold $b$ is a way to cope with the lack of data for estimation of $\left\{\hat{\gamma}_j^2\right\}_{j=k+1}^{n}$ where $k$ is the true order (see the discussion above). In this manner, we first estimate the variances using the constraint $\gamma \geq 0$, and then we impose the threshold $b$ to make sure that no element in $\hat{\gamma}$ is severely underestimated (see Section 7.4, Example 3). We found that this way of imposing thresholds on $\gamma$ gave the best numerical performance.

### 7.3.2 Estimation of $\{\gamma_j^2\}_{j=1}^{n}$ by Penalization

The problem of estimating $\{\gamma_j^2\}_{j=1}^{n}$ using the LS estimate $\hat{h}_{LS,n}$ from a single data realization is ill-posed. Tikhonov regularization [158] is a commonly used method for solving such problems. This regularization consists of an additive penalty term which depends on the parameter of interest (see below). This penalty can be designed to, e.g., shrink the estimate towards zero, or limit its variability. Using the \textit{a priori} assumption that $\{\gamma_j^2\}_{j=1}^{n}$ is a smooth sequence, we impose a penalty on its second order difference (using higher order differences is also possible) and estimate it from

$$\hat{\gamma} = \arg \min_{\gamma \geq 0} \|\gamma - [\hat{h}_{LS,1}]^2 \cdots [\hat{h}_{LS,n}]^2]T\|^2 + \lambda \|L\gamma\|^2.$$  \hspace{1cm} (7.18)
Here,
\[
L = \begin{bmatrix}
1 & -2 & 1 & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 1 & -2 & 1
\end{bmatrix} \in \mathbb{R}^{(n-2)\times n}
\]
is the second order difference matrix and \( \lambda \geq 0 \) is a user parameter which determines the amount of smoothing.\(^2\) Equation (7.18) can be efficiently solved using quadratic programming. \( \lambda \) should be set to a low value if \( \{\gamma_j^2\}_{j=1}^n \) is believed to have large variations, and a high value if \( \{\gamma_j^2\}_{j=1}^n \) is believed to be very smooth (the extreme case \( \lambda = \infty \) forces \( \gamma_j^2 - \gamma_{j-1}^2 \) to be constant with \( j \)). Alternatively, \( \lambda \) can be selected in a fully automatic manner (i.e., without any user parameters) by using the generalized cross-validation (GCV) [52]. An efficient implementation of the GCV for the problem under consideration is given in [60].

7.3.3 Multiple Coefficient Vector Realizations

As mentioned in the beginning of this section, the main difficulty in estimating \( \{\gamma_j^2\}_{j=1}^n \) comes from the fact that only one realization of \( h \) is available. However, if \( m \) data realizations with different coefficient vector realizations \( \{h^{(i)}\}_{i=1}^m \) were available, with the same \( \{\gamma_j^2\}_{j=1}^n \) and \( \sigma^2 \) (the regressor matrices, however, need not be the same) and of the (unknown) orders \( \{k^{(i)}\}_{i=1}^m \), the smoothing approach described in the above two subsections may not be necessary. Assuming \( m \) such realizations of \( h \) are available, then \( \gamma_j^2 \) can be estimated using (7.16), where we replace \( E\|h_j]\|^2 \) by the mean of the squared norms of the full model LS estimates’ ((7.11) with \( k := n \) \( j \)th element:

\[
\hat{\gamma}_j^2 = \frac{1}{m} \sum_{i=1}^m \frac{\|\hat{h}_{LS,j}^{(i)}\|_2^2}{\sum_{k=\max\{j,n_{\min}\}} P(M_k)}, \quad j = 1, \ldots, n. \quad (7.19)
\]

If smoothing or thresholding of the variance profile is desired (e.g., if \( m \) is relatively small), it is straightforward to use the above parameterization or penalization approaches directly on the estimates from (7.19).

Similarly, to estimate \( \sigma^2 \) one can simply average the estimates from (7.12) for the different realizations of \( h \):

\[
\hat{\sigma}^2 = \frac{1}{m} \sum_{i=1}^m \frac{1}{N-n} \|y - X^{(i)} \hat{h}_{LS,n}^{(i)}\|^2
\]

where \( X^{(i)} \) is the regressor matrix corresponding to the \( i \)th realization of \( h \).

\(^2\)By looking at the matrix \((I + \lambda L^T L)^{-1}\) in the closed-form solution of the unconstrained version of (7.18) [147], \( \gamma_{\text{unconstr}} = (I + \lambda L^T L)^{-1}[|\hat{h}_{LS,1}|^2 \cdots |\hat{h}_{LS,n}|^2]^T \), one can get an idea of how \( \lambda \) affects the smoothing.
7.4 Numerical Examples

We evaluate the performances of the methods by means of Monte-Carlo simulations. The performance of BOSS (7.3) is measured in terms of the percentage of correctly selected orders:

\[
\frac{1}{M} \sum_{m=1}^{M} \delta_{\hat{k}^{(m)} - k^{(m)}} \cdot 100 \% 
\]

where \(\delta_t\) denotes the discrete-time unit impulse and \(\hat{k}^{(m)}\), \(k^{(m)}\) denote the estimated and true orders for realization number \(m\), respectively. For the evaluation of BPM (7.9) we use the empirical MSE of the coefficient estimates:

\[
\frac{1}{M} \sum_{m=1}^{M} \|\hat{h}^{(m)} - h^{(m)}\|^2 
\]

where \(\hat{h}^{(m)}\) and \(h^{(m)}\) denote the estimated and true coefficient values, respectively, for realization number \(m\) (if the vectors have different lengths, they can be zero-padded to the same length). \(M\) is the total number of Monte-Carlo runs and we choose \(M = 10^5\).

For each Monte-Carlo trial we generate data from a model \(M_k\) where the order \(k\) is chosen uniformly at random between \(n_{\text{min}} = 1\) and \(n = 30\) (these limits are supplied to the estimators). We set \(N = 50\). The true variance profiles, \(\{\gamma^2_j\}_{j=1}^n\) are constructed from (7.14) where the first column of \(\Psi\) consists of ones, \([1 \cdots 1]^T\), the second is exponentially decaying, \([\nu_1 e^{-0.1} \cdots \nu_1 e^{-0.1n}]^T\), and the third is exponentially increasing, \([\nu_2 e^{0.1} \cdots \nu_2 e^{0.1n}]^T\). The normalization factors \(\nu_1\) and \(\nu_2\) are set such that all columns in \(\Psi\) have a squared \(L_2\)-norm equal to \(n\). The vector \(\alpha\) has independent squared \(N(0,1)\) elements (i.e., each element is generated from a \(\chi^2(1)\)-distribution). Finally, the so generated \(\{\gamma^2_j\}_{j=1}^n\) are normalized such that \(\|\gamma\|^2 = 1\). Naturally, these choices are somewhat arbitrary (as any specific numerical example has to be), but having compared with other examples we believe that it shows a fair comparison of the considered methods.

We consider the following methods: (a) The well-known corrected (for short data sequences) information criterion by Akaike (AICc) [20] for order selection and LS (7.11) with the AICc order for coefficient estimation; (b) The well-known Bayesian information criterion (BIC) [127] for order selection, and LS (7.11) with the BIC order, as well as the BIC model weighted coefficient estimate (see (5.19) in Chapter 5) for estimation of \(h\); (c) BOSS/BPM (7.3)/(7.9) with knowledge of the true \(\{\gamma^2_j\}_{j=1}^n\), \(\sigma^2\); (d) Empirical BOSS/BPM using \(\hat{\sigma}^2\) from (7.12) and estimating \(\{\gamma^2_j\}_{j=1}^n\) as in Section 7.3.1 with the true \(\Psi\); (e) Empirical BOSS/BPM using \(\hat{\sigma}^2\) from (7.12) and estimating \(\{\gamma^2_j\}_{j=1}^n\) as in Section 7.3.1 with the true \(\Psi\).

\[^{3}\text{This means that } \alpha \text{ has positive elements. The constraint } \alpha \geq 0 \text{ is stronger than the constraint } \gamma = \Psi \alpha \geq 0 \text{ used in (7.17)-(7.18). (Note that } \alpha \geq 0 \text{ implies that } \gamma = \Psi \alpha \geq 0 \text{ for the } \Psi \text{ we consider.) However, we choose not to exploit the prior knowledge that } \alpha \geq 0 \text{ when using our empirical Bayesian methods described in Section 7.3, even if doing so would likely increase their performance. The reason for not exploiting this information is that we want to avoid tailoring the data in our numerical examples too much to our methods.}\]

7.4. Numerical Examples

7.4.1 Example 1: Uncorrelated Regressors

First we consider an example in which all elements of $\mathbf{X}$ are i.i.d $\mathcal{N}(0,1)$. In Figure 7.1 we study the model order selection performances and in Figure 7.2 we study the coefficient estimation performances. We observe that the empirical BOSS/BPM methods give a better performance than $\text{AIC}_c$ and BIC do (generally about 0.5 to 1 dB better, and sometimes significantly more). Also, the differences between the different empirical Bayesian methods are very small. When estimating $\sigma^2$ and $\{\gamma_j^2\}_{j=1}^n$ the model selection performance drops about 1 dB below that of the optimal BOSS where these parameters are known. The

---

$^4$ $b_f = 0.01$, $b_f = 0.05$ and $b_f = 0.3$ gave results similar to those shown here.
Figure 7.2: Coefficient estimation performance for uncorrelated regressors. (The small plot is a closeup.)
7.4. Numerical Examples

Figure 7.3: Estimation of $\gamma$ for two sample data realizations in Example 1 with $\sigma^2 = -10$ dB (plotted as lines with squares and lines with x-marks, respectively). The true order for the first data realization (lines with squares) was $k = 8$, and for the second data realization (lines with x-marks) the true order was $k = 20$.

The corresponding loss in coefficient estimation performance for empirical BPM is about 0.3 dB.

For this example we also show, in Figure 7.3, two representative realizations of the coefficient variances $\{\gamma_j^2\}_{j=1}^n$ together with their estimates, obtained using the approaches in Section 7.3 when $\sigma^2 = -10$ dB. As expected, the estimated variances sometimes deviate rather much from their true values (especially for indices larger than the true order $k$), which reduces performance as compared to BOSS/BPM using the true variances. However, some features of the variance profiles are captured by the variance estimates, and this is enough for the empirical BOSS/BPM methods to outperform AIC/BIC.

7.4.2 Example 2: Correlated Regressors

We now let the regressors be correlated, such that $x_i = (z + v_i)/\sqrt{2}$ where $z$ and $\{v_i\}_{i=1}^n$ are i.i.d. $\mathcal{N}(0, I)$ vectors. Thus $E\{x_i x_j^H\} = \frac{1+\delta_{i,j}}{2}I$. In Figure 7.4 we study the model order selection performances and in Figure 7.5 we study the coefficient estimation performances. Also for this example, the empirical versions of BOSS and BPM are able to improve upon the BIC and AIC$_c$. 
Figure 7.4: Model order selection performance for correlated regressors. (The small plot is a closeup.)
Figure 7.5: Coefficient estimation performance for correlated regressors. (The small plot is a closeup.)
The different empirical Bayesian methods show similar performance: for model selection, about 1 dB below the optimal BOSS, which exploits full knowledge of \( \sigma^2 \) and \( \{\gamma_j^2\}_{j=1}^n \), and for coefficient estimation about 0.1 to 0.4 dB below the optimal BPM. The BIC model weighting approach (5.19) for coefficient estimation improves with about 0.2 dB upon LS using the BIC model order.

### 7.4.3 Example 3: Variance Profile Mismatch

We also study how BOSS and BPM are affected by mismatch in the variance profile \( \gamma \). This is interesting because, as we discussed in Section 7.3 and as can be seen in Figure 7.3, \( \gamma \) is difficult to estimate accurately.

The data are generated as in Example 1 (with uncorrelated regressors). However, this time we supply BOSS and BPM with mismatched \( \gamma \)-values such that \( \gamma_{\text{used}} = c \gamma_{\text{true}} \), where \( c \) is varied. The empirical Bayesian variants, included for comparison, estimate \( \gamma \) from the data and are thus unaffected by \( c \). The results for order selection are shown in Figure 7.6. The coefficient estimation results are similar, and are not shown here. We remark upon the fact that the performances of BOSS/BPM are rather robust to mismatching \( \gamma \)-values. Furthermore, they seem less sensitive to the use of too large values for \( \gamma \) than the use of too small \( \gamma \)-values. This observation motivates the use of the threshold \( b \) as a “robustification” parameter, which serves the purpose of avoiding (harmful) severely underestimated values in \( \gamma \).
7.4.4 Example 4: Multiple Coefficient Vector Realizations

Finally, we consider the same setting as in Example 1, but this time with 10 coefficient vector realizations for each $\gamma$. We use the methodology described in Section 7.3.3 to compute the empirical Bayes BOSS and BPM. In Figure 7.7 we show the model selection results. Compared to Example 1, the empirical BOSS methods have improved—their performance is now very similar to that of the optimal BOSS for low noise levels. The corresponding coefficient estimation plot is omitted for brevity, but is relatively similar to Figure 7.2. The empirical BPM now performs less than 0.1 dB below the optimal BPM method.

7.5 Conclusions

We have studied linear regression with an unknown model order assuming zero-mean Gaussian noise and a zero-mean Gaussian distributed coefficient vector. Under this model we have derived empirical Bayesian versions of the maximum a posteriori probability model order selector and the MMSE coefficient vector estimate. These empirical Bayesian methods have been shown to outperform the classical approaches $\text{AIC}_c$ and BIC, both in model order selection and co-
efficient vector estimation examples. Results similar to those shown have also
been obtained from many other numerical examples which have to be omitted
here due to space constraints. Since our methods have relatively low compu-
tational complexity and show good performance, we consider them attractive
alternatives in the context of the model (7.2).

At www.it.uu.se/katalog/ys/software/empBOSS_BPM we have posted a
free software implementation (in Matlab) of the methods presented here.
Part II

Sparse Modeling
Chapter 8

A Model Averaging Approach for Sparse Communication Channel Equalization

When performing equalization in communication applications it is important to use a good estimate of the channel over which the signal was transmitted. In this chapter we consider the concept of model averaging, where several channel estimates are used in a weighted manner for equalization of sparse communication channels with intersymbol interference. We show via a numerical example that the bit-error-rate (BER) can be reduced by our suggested method, compared to the BERs obtained using channel estimation methods which do not assume any sparsity of the channel.

8.1 Introduction

This chapter is concerned with estimation of sparse linear frequency-selective finite impulse response (FIR) channels which are to be used for equalization. This is an important problem, since when equalization is performed, good knowledge of the channel is essential for low error rates.

A sparse channel is a channel for which many taps are zero-valued. For a few examples of where such channels can appear, see [65]. Sparse channels have previously been studied, e.g., in [65; 152], where methods to detect non-zero channel taps were derived. In this chapter we suggest an approach similar to that in [65], but extended to a model averaging framework.
For some time, in the statistical literature, model averaging [163] has been considered a sound way of improving the quality of inference. The model averaging approach does, however, not seem common in the communication theory literature, where often only one model is chosen and the remaining ones are ignored (some exceptions include [65; 137; 151]). While this might work well in many cases, it is sub-optimal [64]. This can be intuitively realized by considering a case for which there are many candidate models which seem equally probable according to some criterion. Basing all inferences on one of these models and ignoring the others means that the information we have on the model uncertainty is discarded. By considering several or all candidate models in a weighted manner, more information is exploited and we can expect better performance. Model averaging for time-dispersive communication channels, which we consider here and will denote channel averaging from here on, was previously studied for non-sparse channels in [87] (see also the Chapters 5-7 in this thesis). In this chapter we consider channel averaging for sparse channels. While channel averaging is not optimal from a decision theoretic point of view, equalization via an averaged channel estimate can be seen as a pragmatic approximation to an optimal symbol sequence detector (cf. Chapter 6).

In the next section we present some methods previously considered for channel estimation and extend them to a model averaging approach for sparse channels. In Section 8.3, we show a numerical example which illustrates the performance obtained by an equalizer using our averaged channel estimate, and compares it to some other approaches. Finally, in the closing Section 8.4, we discuss our results.

### 8.2 Sparse Channel Averaging

We consider channel averaging based on the Information Criterion (IC) approach; see Chapter 2. This channel averaging approach consists of the estimation of several communication channels, which then are added together in a weighted manner. This results in an averaged channel which can be used for
8.2. Sparse Channel Averaging

equalization. Let

\[ y = [y(0) \cdots y(N-1)]^T \in \mathbb{C}^N, \text{ the received data vector} \]

\[ y_L = [y(L) \cdots y(N-1)]^T \in \mathbb{C}^{N-L}, \text{ the truncated data} \]

\[ h_L^{(n)} \in \mathbb{C}^{L+1} = [h_0^{(n)} \cdots h_L^{(n)}]^T, \text{ the impulse response of a sparse} \]

channel with a maximum order \( L \) and \( n \leq L + 1 \) non-zero taps

\[ p_n(y_L|h_L^{(n)}) = \text{the likelihood function associated with a channel with impulse response } h_L^{(n)} \]

\[ \hat{h}_L^{(n)} = \text{arg max}_{h_L^{(n)}} p_n(y_L|h_L^{(n)}), \text{ the Maximum Likelihood (ML)} \]

\[ \text{estimate of } h_L^{(n)}. \]

Our problem of interest is to estimate the unknown channel \( h_L \) (which we assume to have a maximum order of \( L \)) via the transmission of a known sequence of training symbols, \( \{s(t)\}_{t=0}^{N-1} \) (chosen from a finite alphabet), over the channel. The channel estimate can then be used in an equalizer to detect unknown information symbols; see the numerical example in Section 8.3. The data collected at the receiver take on the form\(^2\)

\[ y(t) = \sum_{l=0}^{L} s(t-l)h_l + e(t) = s_L^T(t)h_L + e(t), \quad t = 0, \ldots, N-1 \quad (8.1) \]

where \( s_L^T(t) = [s(t) \cdots s(t-L)] \), and \( e(t) \) is a noise term which we assume to be zero mean, white and circularly Gaussian with variance \( \sigma^2 \). We also assume that the unknown channel \( h_L \) is sparse, i.e., “many” of the elements in \( h_L \) are zero. We further assume that the known training signal \( s = [s(0) \cdots s(N-1)]^T \) has been chosen to be white (or have an impulse-like autocorrelation sequence). Note that this is to be preferred for channel estimation purposes, as a white \( s \) is optimal when the channel is unknown [11].

To be able to form a good estimate \( \hat{h}_L^{(n)} \) of a sparse channel we should know the appropriate channel length \( L \), the appropriate number \( n \) of non-zero taps, and also which elements in \( \hat{h}_L^{(n)} \) should be non-zero. These pieces of information will be denoted the structure of the channel. The problem of finding appropriate channel structures can be treated in many different ways, e.g., by hypothesis testing or by using IC [140] (see also Chapter 4). In this chapter we present a computationally efficient way of obtaining several sparse candidate channels which can be averaged and used for equalization.

\(^1\)By “order” we mean the largest value \( m \leq L \) for which \( h_m^{(n)} \) is non-zero.

\(^2\)Note that for values of \( t \) lower than the channel order, (8.1) requires knowledge of (the unavailable) \( s(k) \) for \( k < 0 \). While we could, e.g., set \( s(k) = 0 \) for \( k < 0 \) we instead choose to not use the received data samples in \( y \) which require these unavailable training symbols (see (8.3), (8.4), (8.9), (8.10) where we use the truncated data vector \( y_L \) instead of \( y \)).
8.2.1 Information Criteria for Selecting a Single Channel

The Generalized IC (GIC) can, for complex valued data, be written in the following form (see Chapter 2):

\[ \text{GIC}(\hat{h}_L^{(n)}) = -2 \ln p_n(y_L|\hat{h}_L^{(n)}) + 2n\nu \]  

(8.2)

where \( n \) is the number of non-zero taps in \( \hat{h}_L^{(n)} \). The first term of the GIC shows how well the known sequence of symbols convolved with the channel estimate fits the received data. For a linear channel in complex Gaussian noise, we have

\[ -2 \ln p_n(y_L|\hat{h}_L^{(n)}) = 2(N - L) \ln \tilde{\sigma}_n^2, \]  

(8.3)

where

\[ \tilde{\sigma}_n^2 = \frac{1}{N - L} \sum_{t=L}^{N-1} \left| y(t) - s_T^T(t)\hat{h}_L^{(n)} \right|^2. \]  

(8.4)

Note that (8.3) and (8.4) are different from the common form for linear models in Gaussian noise [150]. This is so because we use only a part of the signal \( y \), namely the truncated vector \( y_L \) containing \( N - L \) samples.

For the two well-known AIC [6] (Akaike IC) and BIC [127] (Bayesian IC) we have

\[
\begin{align*}
\text{AIC:} & \quad \nu = 2 \\
\text{BIC:} & \quad \nu = \ln[2(N - L)]
\end{align*}
\]

(8.5a, 8.5b)

(for the complex valued data case). The conventional way of using GIC for channel estimation is to, for some pre-specified interval \( \mathcal{I} \) of the order \( m \), select the single non-sparse channel

\[ \hat{h} = \arg \min_{m \in \mathcal{I}} \text{GIC}(\hat{h}^{(m)}) \]  

(8.6)

where \( \hat{h}^{(m)} \) is the non-sparse channel estimate of order \( m \) (for which the \( m + 1 \) elements \( \{\hat{h}_j\}_{j=0}^m \) are non-zero), see Chapter 2.

8.2.2 Information Criteria for Channel Averaging

In what follows, we will use BIC. The key to using BIC for channel averaging is the observation that the posterior probability of a channel structure \( \mathcal{M}_L^{(n)} \) (which defines the non-zero elements of \( \hat{h}_L^{(n)} \), i.e., the probability of \( \mathcal{M}_L^{(n)} \) given the data vector \( y_L \), can be estimated directly using BIC in the following manner (see Chapter 5):

\[ \hat{P}(\mathcal{M}_L^{(n)}|y_L) = \frac{e^{-\frac{1}{2} \text{BIC}(\hat{h}_L^{(n)})}}{\sum_{\hat{h} \in \mathcal{H}} e^{-\frac{1}{2} \text{BIC}(\hat{h})}} \]  

(8.7)

where \( \mathcal{H} \) is the set of all channels considered, including \( \hat{h}_L^{(n)} \), such that the total probability of the channels in \( \mathcal{H} \) is 1. By using these posterior probability estimates as weights we can form a weighted sum over all channels considered to obtain an averaged channel \( \hat{h}_L^{(n)} \) (see below).
8.2. Sparse Channel Averaging

8.2.3 Information Criteria for Sparse Channel Averaging

Herein we present a computationally efficient way of obtaining sparse channel structures via the estimation of $L+1$ channels with only one non-zero tap.

The assumption that the known input symbols $\{s(t)\}$ are uncorrelated means that the contribution of each tap to the measured signal $y$ is decoupled from that of the other taps. In other words, the taps of a channel can be estimated in a one-by-one fashion, without having to consider the other taps in the channel. This is most easily realized by considering the expected squared difference between the measured signal $y(t)$ and the output $\hat{y}(t)$ of a channel estimate $\hat{h}_L^{(n)}$:

$$E\{|y(t) - \hat{y}(t)|^2\} = E\{|s_L^T(t)(h_L - \hat{h}_L^{(n)}) + e(t)|^2\} = \sum_{l=0}^{L} |h_l - \hat{h}_l^{(n)}|^2 \sigma_s^2 + \sigma_e^2$$  \hspace{1cm} (8.8)

where $\sigma_s^2$ denotes the variance of the training signal $s$. The lack of cross-terms between the estimated taps $\{\hat{h}_l^{(n)}\}$ in (8.8) implies that we can consider each tap separately and rank them in order of importance (see [65]).

Consider the channel $h_L^{(1)} = [h_0 \cdots h_L]^T$ which has $h_l$ as its single non-zero tap. By estimating $h_l$ for $l = 0, \ldots, L$ via Least Squares (LS) (which is identical to ML under the Gaussian white noise assumption [140; 150]),

$$\hat{h}_l = \frac{\sum_{t=1}^{N-1} s^*(t-l)y(t)}{\sum_{t=1}^{N-1} |s(t-l)|^2}, \quad l = 0, \ldots, L$$  \hspace{1cm} (8.9)

where $(\cdot)^*$ denotes the complex conjugate, we obtain $L+1$ one-tap ML channel estimates. Using these channel estimates we can compute the corresponding residual variances

$$\hat{\sigma}_l^2 = \frac{1}{N-l} \sum_{t=l}^{N-1} |y(t) - s(t-l)\hat{h}_l|^2, \quad l = 0, \ldots, L$$  \hspace{1cm} (8.10)

and obtain a channel structure $M_L^{(n)}$ for each number of non-zero taps $n$ from $n = 1$ up to a user selected limit $n = \bar{n} \leq L + 1$, such that the positions of the non-zero taps of $\hat{h}_L^{(n)}$ correspond to the $n$ smallest values of $\{\hat{\sigma}_l^2\}_{l=0}^{L}$. This amounts, in some sense, to choosing the $n$ taps most important for describing the received signal (cf. (8.8)). Once a channel structure for each $n = 1, \ldots, \bar{n}$ has been decided upon, we can find the corresponding ML/LS estimates $\{\hat{h}_L^{(n)}\}_{n=1}^{\bar{n}}$ and average them using (8.2), (8.5b), and (8.7) such that the averaged channel estimate $\hat{h}_L^{(\bar{n})}$ is given by

$$\hat{h}_L^{(\bar{n})} = \sum_{n=1}^{\bar{n}} \hat{P}(M_L^{(n)}|y_L)\hat{h}_L^{(n)}$$  \hspace{1cm} (8.11)

(see Chapter 5). This averaged estimate of the channel can then be used for channel equalization.
8.3 Numerical Example

We illustrate the error performance of various channel estimators by means of Monte-Carlo simulations.

First we use the various methods to estimate the channel via the transmission of a known training sequence, and then we use the obtained channel estimates in a Viterbi equalizer [114] to detect unknown sequences of symbols transmitted over the channel. The Viterbi equalizer detects unknown symbols \( \{ \tilde{s}(t) \} \) by minimizing

\[
\sum_t \left| \tilde{y}(t) - \sum_k \hat{h}_k \tilde{s}(t - k) \right|^2
\]

with respect to \( \{ \tilde{s}(t) \} \), where \( \{ \hat{h}_k \} \) denotes the taps of the channel estimate, and \( \{ \tilde{y}(t) \} \) denotes the data received during the transmission of \( \{ \tilde{s}(t) \} \). The above is equivalent to the ML sequence detector assuming white Gaussian noise.

The performance is illustrated by the bit-error-rates (BERs) of the equalized signals.

8.3.1 Channel Model

We consider random sparse Rayleigh fading channels (8.1) with maximum order \( L = 7 \), i.e., with \( L + 1 = 8 \) filter taps (some of which might be zero). The channels are generated using exponentially decaying delay profiles constructed according to, for \( l = 0, \ldots, L \),

\[
d_l = \begin{cases} 
  e^{-0.25t} & \text{if } 0.5 > P \in U(0,1) \\
  0 & \text{else}
\end{cases}
\]

where \( U(0,1) \) denotes the uniform distribution between 0 and 1. The channels \( h_L \) have independent, zero mean, complex Gaussian coefficients with variances

\[
E\{ |h_0|^2 |h_1|^2 \cdots |h_L|^2 \} = \frac{[d_0 \ d_1 \ \cdots \ d_L]}{\sum_{l=0}^{L} d_l}.
\]

The scaling factor in the denominator ensures that all delay profiles have unit energy. As the known training input sequence we use a pseudorandom \( N = 26 \) sample BPSK sequence \( s = [s(0) \cdots s(N-1)]^T = [-1, -1, 1, -1, 1, 1, 1, -1, -1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1, 1, -1, 1]^T \) used by the GSM system [35]. (In Figure 8.1 we study the “whiteness” of \( s \) by showing its autocorrelation sequence [147].) The variance \( \sigma^2 \) of the circular Gaussian noise \( e(t) \) is varied to obtain different SNRs (Signal-to-Noise Ratios).

In Figure 8.2 we show the BERs corresponding to a Viterbi equalizer using the estimated channels to detect unknown randomly generated 170 sample long BPSK symbol sequences. 10 000 Monte Carlo simulations were run for several SNRs. The following channels estimates were used: a) non-sparse channel with 7 non-zero taps, estimated via ML [87]; b) non-sparse channel with 8 non-zero taps, estimated via ML [87]; c) non-sparse ML channel estimate where the
Figure 8.1: The autocorrelation sequence [147] of the \( N = 26 \) sample long BPSK training sequence \( s \) used in the numerical example.
channel length was estimated using BIC ((8.5b), (8.6) with $I = \{0, \ldots, 7\}$); d) non-sparse BIC channel average using non-sparse ML channel estimates of increasing lengths ($k_{m+1}^{(m+1)}$ for $m = 0, \ldots, 7$) and (8.11) modified in a straightforward manner (see also [151]); e) the sparse channel averaging approach (8.11) with $\bar{n} = 8$; and f) coherent detection, where the true channel was used for equalization.

We observe that an improvement of about 1 dB is obtained when using our sparse channel averaging approach instead of the standard BIC or the non-sparse channel averaging. The non-sparse channel with 8 non-zero taps also performs rather well in this example, which is because it hits the correct channel order in as many 50% of the cases (see (8.12)) and never uses an order which is too low. Had the example been constructed such that the true channel order was more uncertain, the BIC channel estimate and the non-sparse channel averaging would likely perform better than this fixed-order channel estimate [87].

8.4 Conclusions

We have introduced a channel averaging approach for sparse channels and demonstrated its ability to outperform conventional single channel and non-sparse channel averaging approaches for channel equalization (see Section 8.3).
The sparse channel averaging approach is only slightly more computationally complex than a standard channel selection approach. Therefore sparse channel averaging appears to be an attractive way of decreasing the error rates when the communication channels are expected to be sparse.
Linear Regression With a Sparse Parameter Vector

We consider linear regression under a model where the parameter vector is known to be sparse. Using a Bayesian framework, we derive the minimum mean-square error (MMSE) estimate of the parameter vector, and a computationally efficient approximation of it. We also derive an empirical-Bayesian version of the estimator, which does not need any a priori information, nor does it need the selection of any user parameters. As a byproduct, we obtain a powerful model (“basis”) selection tool for sparse models. The performance and robustness of our new estimators are illustrated via numerical examples.

9.1 Introduction

9.1.1 Problem Formulation

Consider the linear regression model

\[ y = Xh + e \quad (9.1) \]

where \( h \in \mathbb{R}^n \) is a parameter vector, \( y \in \mathbb{R}^N \) is a vector of observations, \( X \in \mathbb{R}^{N \times n} \) is a known regressor matrix, and \( e \in \mathbb{R}^N \) is a vector of noise. The task is to estimate \( h \), given that \( y \) was observed.

As is well-known, the least-squares (LS) estimate of \( h \) is given by the minimizer of \( \|y - Xh\|^2 \) with respect to \( h \), i.e., (see [75], for example):

\[ \hat{h}_{\text{ls}} = (X^T X)^{-1} X^T y. \quad (9.2) \]
provided $N \geq n$ and $X$ is full rank. If the noise is zero-mean, white and Gaussian, then the LS estimate coincides with the maximum-likelihood (ML) estimate [75]. Hereafter, we shall assume that $e \sim \mathcal{N}(0, \sigma^2 I)$.

The LS estimate is commonly used owing to its simplicity and its connection to ML. However, if something is known about $h$ a priori (before the data are collected), then one can do better than the LS estimate. For example, if one knows that $h \sim \mathcal{N}(0, \gamma^2 I)$ a priori, then the estimate of $h$ which has the smallest mean-square error (MSE, $E[\|h - \hat{h}\|^2]$) is given by the conditional mean of $h$ given that $y$ was observed [75]:

$$
\hat{h}_{\text{MMSE}} = E\{h|y\} = \left( \frac{\sigma^2 I}{\gamma^2} + X^T X \right)^{-1} X^T y.
$$

(9.3)

Note that when $\gamma^2 \to \infty$, corresponding to the observer having no a priori knowledge of $h$, then the MMSE and LS estimates coincide.

Generally, the minimum MSE (MMSE) estimate is better (in the MSE sense) than the LS estimate, owing to the influence of the a priori knowledge of $h$. We now ask the question: If $h$ is known to be sparse, that is, some elements of $h$ are likely to be equal to zero, can we do even better than the MMSE estimate (9.3)? And if so, how much better can we do?

9.1.2 Examples of Applications for Sparse Models

Sparse linear models are relevant in a variety of applications. For example, in a statistical data analysis one may know before the measurement that the data are likely to be explained by only a few factors. Another example is the estimation of communication channel impulse responses where the bandwidth is so large that individual multipath components can be resolved, and are separated by more than one sample period. In practice this may occur in indoor ultrawideband (UWB) radio systems [168], or for underwater communications [78]. Yet two instances where sparse modeling can be applied are in wavelet theory [71; 167], and spatial signal processing for source localization [94]. The introductory sections of [65; 166] contain many other examples along with references to the pertinent literature.

9.1.3 Related Work on Regression with Sparse Models

Linear regression for sparse models has been studied both in the statistics community and in the signal processing literature. We next review some of the representative contributions and approaches we are aware of.

1. A large class of methods [22; 116; 117] is based on Bayesian maximum a posteriori (MAP) estimation of $h$ assuming a prior density for $h$ which

\[ \text{Unlike the LS estimate, however, the MMSE estimate is biased.} \]
induces sparsity. A typical such prior is the following:

$$p(h) \sim \exp \left( -\frac{b}{2} \sum_{j=0}^{n-1} |h_j|^\rho \right)$$

(9.4)

where \(\rho\) and \(b\) are user parameters which satisfy \(0 < \rho \leq 1\) and \(b > 0\). With Gaussian noise, the MAP estimate follows as

$$\hat{h} = \arg\max_h p(h|y) = \arg\max_h \left( p(y|h)p(h) \right)$$

$$= \arg\min_h \left( \frac{1}{\sigma^2} \|y - Xh\|^2 + b \sum_{j=0}^{n-1} |h_j|^\rho \right).$$

(9.5)

One can interpret the resulting estimate as an LS fit with a penalty term. Typically, the estimates obtained in this way have a sparse structure. The parameter \(b\) balances the conflicting objectives of minimizing the LS fit residual (this calls for a small value of \(b\)) and obtaining a sparse estimate (this calls for a large \(b\)). Note that solving (9.5) can be computationally hard.

2. The “Lasso” (Least absolute shrinkage and selection operator) method [157] estimates \(h\) by minimizing the LS criterion \(\|y - Xh\|^2\) subject to a \(L_1\)-norm constraint on the parameter vector. More precisely, Lasso finds \(h\) via

$$\hat{h}_{\text{Lasso}} = \arg\min_h \|y - Xh\|^2 \quad \text{subject to} \sum_{j=0}^{n-1} |h_j| \leq c$$

(9.6)

where \(c\) is a user parameter. Note that (9.6) can be cast as a linearly constrained quadratic problem, which can be efficiently solved.

Interestingly, Lasso has a Bayesian interpretation. Namely, the Lasso estimate of \(h\) turns out to be the same as the MAP estimate obtained if \(h\) has a prior density of the form

$$p(h) = \left( \frac{1}{2\pi} \right)^n \exp \left( -\frac{1}{2} \sum_{j=0}^{n-1} |h_j|^2 \right).$$

(See Section 5 of [157], and the references therein.) From this perspective, Lasso is related to the methods of [22; 116; 117].

Using a small enough value for \(c\) typically leads to a parameter vector estimate for which many coefficients actually are equal to zero (i.e., not only “small”). Explanations for why this happens are given in Section 2 of [157]. To choose \(c\), one can take one of two possible standpoints: either

---

2Hereafter, \(h_j\) stands for the \(j\)th element of the vector \(h\), and “\(\sim\)” stands for equality up to irrelevant constants.
one sticks to the Bayesian interpretation and sets $c$ to the best \textit{a priori} belief one has about the data. Alternatively, one attempts to “estimate” $c$ from the data in some way, in an effort to obtain a method which is completely free of user parameters. For the latter task, [157] suggested two algorithms based on cross-validation and Stein’s unbiased estimate of risk, respectively.

There also exist other more recent methods closely related to Lasso, such as Forward stagewise regression and “Lars” (Least angle regression) [34].

3. The SBL (Sparse Bayesian Learning) method of [166] is based on the assumption that $h_j$ are independent, zero-mean Gaussian with unknown variances $\gamma_0^2, \ldots, \gamma_{n-1}^2$. Under this assumption, the likelihood function for $y$ is

$$ p(y|\sigma^2, \gamma_0^2, \ldots, \gamma_{n-1}^2) = \int_{\mathbb{R}^n} p(y|h, \sigma^2)p(h|\gamma_0^2, \ldots, \gamma_{n-1}^2)dh $$

$$ = \frac{1}{(2\pi\sigma^2)^{n/2}(\prod_{j=0}^{n-1} \gamma_j^2)^{1/2}} \int_{\mathbb{R}^n} \exp \left( -\frac{1}{2\sigma^2} \|y - Xh\|^2 - \sum_{j=0}^{n-1} \frac{h_j^2}{\gamma_j^2} \right) dh. $$

The variances $\gamma_0^2, \ldots, \gamma_{n-1}^2$ and $\sigma^2$ are treated as hyperparameters and can be eliminated from $p(y|\sigma^2, \gamma_0^2, \ldots, \gamma_{n-1}^2)$ by maximizing it using the expectation-maximization (EM) algorithm (see Appendix A). During the execution of the EM algorithm some variances $\gamma_j^2$ are driven to zero, which results in a sparse estimate of $h$. One fundamental difference between [166] and the other methods discussed in this paper (including our proposed algorithm), is that in [166] the number of hyperparameters (which are eventually eliminated by maximizing the likelihood function) grows linearly with the dimension of $h$.

4. In [102] the authors suggested to estimate $h$ via

$$ \hat{h}_{\rho\text{-norm}} = \arg\min_h (\|y - Xh\|_\rho + \lambda\|h\|_\rho) \quad (9.7) $$

where $\lambda$ and $\rho$ are user parameters. The user parameter $\lambda$ balances the conflicting objectives of minimizing the residual and obtaining an estimate with a sparse structure, in the same manner as the parameter $b$ does in (9.5). The estimate in (9.7) can be interpreted as the Bayesian MAP estimate if one assumes that $e$ and $h$ are mutually independent with densities $p(h) \equiv \exp(-c_h\|h\|_\rho)$ and $p(e) \equiv \exp(-c_e\|e\|_\rho)$, where $\lambda = c_h/c_e$. Note that [102] is similar in spirit to [22; 116; 117], however in its Bayesian interpretation [102] departs from the assumption of Gaussian noise.

\footnote{Here, $\|\cdot\|_\rho$ stands for the $L_\rho$-norm. (The special case of $\rho = 2$ gives the Euclidean norm.)}
5. The paper [65] focuses on the recursive ("on-line") identification of finite-impulse-response (FIR) filters. The method described therein is based on the least mean-squares (LMS) algorithm, with the addition of a technique that determines which coefficients in the filter are “active”. The paper offers strong theoretical arguments on consistency of the selection technique. However, since it is less closely related to our work than the other techniques reviewed above, we omit a more detailed explanation.

All methods (including the standard MMSE estimate (9.3)) which use an explicit prior for $h$ allow the regression problem to be overcomplete, that is $n > N$. For $n > N$ one can think of the use of priors on $h$ as a way of regularizing the problem (the estimation of $h$ from $y$ with $n > N$ without a prior would be ill-posed).

Most of the methods reviewed above are Bayesian (or at least they have a Bayesian interpretation) and as such they are arguably optimal (in the sense of MAP) if the model and the a priori knowledge assumed in the algorithm match perfectly with the process that generates the data. However, none of the above methods has any clear connections to the sparseness structure of the model in terms of the probability of a given coefficient being equal to zero. The goal of this paper is to present an estimator for which this connection is explicit.

Note that many existing methods are primarily concerned with determining the structure of $h$ (i.e., finding out what elements are zero). This problem is often called model, or “basis”, selection. The objective is then to find a (small) set of “basis vectors” (i.e., columns of $X$) such that the observed vector $y$ can be expressed as a linear combination of these basis vectors. In this paper our primary objective is to estimate $h$ from $y$ as accurately as possible, although we will obtain a solution to the model selection problem as well. Note that accurate estimation of $h$ is often of interest. One notable application is channel estimation in communications, where the communication performance frequently is directly related to the quality of the channel estimates.

9.1.4 Contribution of This Work

We propose a computationally efficient method for computing the MMSE estimate of the parameter vector $h$, under the explicit model assumption that a given coefficient $h_j$ is equal to zero with a certain probability. Our method is Bayesian, and as such it requires certain a priori assumptions (how easy these are to justify, depends on the context). Specifically, we require knowledge of the noise variance, the variance of the coefficients, and the probability that a given coefficient is nonzero (that is, the probability that a given regressor is active). The a priori parameters, to be described in Section 9.2, required by the algorithm have clear and unambiguous interpretations and they are explicit in the estimator and its derivation; also, by varying them one can directly study how the estimate is affected. It turns out that the estimate is relatively insensitive to the values of the a priori parameters. Encouraged by this observation, we devise an empirical-Bayesian variant of the estimator which autonomously determines the a priori parameters from the data. The resulting algorithm is
useful in the event that the user has no a priori knowledge of the data at all, and hence wants to avoid selecting user parameters. Finally, we show how a powerful model selection tool emerges as a byproduct from our development as well.

9.2 Model

We shall assume that $h$ has a sparse structure which can be described as a mixture of $2^n$ components, $H_0, ..., H_{2^n-1}$. That is, more explicitly,

$$p(h) = \sum_{i=0}^{2^n-1} p(h | H_i) P(H_i).$$

For each mixture component $H_i$, a subset of the coefficients of $h$ is constrained to zero and the other coefficients are i.i.d. random variables. Specifically,

$$H_0 : h_0, ..., h_{n-1} \text{ i.i.d. } \mathcal{N}(0, \gamma^2)$$

$$H_1 : h_0 = 0; h_1, ..., h_{n-1} \text{ i.i.d. } \mathcal{N}(0, \gamma^2)$$

$$H_2 : h_1 = 0; h_0, h_2, ..., h_{n-1} \text{ i.i.d. } \mathcal{N}(0, \gamma^2)$$

$$...$$

$$H_n : h_{n-1} = 0; h_0, ..., h_{n-2} \text{ i.i.d. } \mathcal{N}(0, \gamma^2)$$

$$H_{n+1} : h_0 = h_1 = 0; h_2, ..., h_{n-1} \text{ i.i.d. } \mathcal{N}(0, \gamma^2)$$

$$H_{n+2} : h_0 = h_2 = 0; h_1, h_3, ..., h_{n-1} \text{ i.i.d. } \mathcal{N}(0, \gamma^2)$$

$$...$$

$$H_{2^n-1} : h_0 = ... = h_{n-1} = 0.$$ (9.8)

Each mixture component has an associated probability $P(H_i)$. Naturally, these probabilities sum to one:

$$\sum_{i=0}^{2^n-1} P(H_i) = 1.$$

The probabilities $\{P(H_i)\}$, as well as $\{\gamma^2, \sigma^2\}$, are assumed to be known, or at least they are set to something sensible. Typically, like in all Bayesian inference, the priors are set to the best “belief” one has before the data are observed. (This is no more strange than assuming that the signal-to-noise ratio is known when performing “classical” MMSE estimation.)

If the coefficients of $h$ are independent and equal to zero with probability $p$ (an assumption which seems reasonable to make in practice, but which is not necessary for the analysis to come), then we have that

$$P(H_0) = (1 - p)^n$$

$$P(H_1) = p(1 - p)^{n-1}$$

$$P(H_{n+1}) = p^2(1 - p)^{n-2}$$

$$...$$

$$P(H_{2^n-1}) = p^n.$$ (9.9)
In this case, \( h_j \) are i.i.d. and drawn from the following mixture density:

\[
p(h_j) = p\delta(h_j) + \frac{1 - p}{\sqrt{2\pi}\gamma^2} \exp \left(-\frac{h_j^2}{2\gamma^2}\right).
\] (9.10)

The data model presented here, also known as the Bernoulli-Gaussian model [79], has an intuitive appeal, and the parameters \( \{\gamma^2, \sigma^2, p\} \) have clear physical interpretations. In particular, the model is a natural description of the situation where the coefficients are Gaussian \textit{a priori}, but it is uncertain whether a given regressor is “active” or not. The use of Gaussian priors on the regressor coefficients has a long history [75], and many physical processes can be well described via mixture densities of the form (9.10). For example, in [128] it was indicated that the coefficients of the sampled impulse response for the indoor radio channel model [123] behaved according to (9.10).

### 9.3 MMSE Parameter Estimation

#### 9.3.1 Derivation of the Estimate

The task we want to tackle is that of computing the MMSE estimate of \( \hat{h} \), given \( y \), under the mixture model (9.8). This estimate is equal to the conditional mean, \( \hat{h}_{\text{MMSE}} = E[h|y] \). We have

\[
\hat{h}_{\text{MMSE}} = E[h|y] = \sum_{i=0}^{2^n-1} P(H_i|y)E[h|y,H_i].
\] (9.11)

In principle, the sum in (9.11) can be computed (assuming we can calculate its terms). The difficulty is that for large \( n \), the number of terms can be unbearably large, and the computational complexity may become unreasonable. Indeed, the computational burden of directly evaluating (9.11) grows exponentially with \( n \).

In order to approximate (9.11) to obtain a computationally feasible expression, we note that \( \hat{h}_{\text{MMSE}} \) can be thought of as a weighted sum of \( E[h|y,H_i] \), where the non-negative weights \( P(H_i|y) \) sum up to one. Furthermore, the terms \( E[h|y,H_i] \) should be of the same order of magnitude, at least for the values of \( i \) for which \( P(H_i|y) \) is significantly different from zero. Therefore, the weighted sum is dominated by the terms for which \( P(H_i|y) \) are large, and a good approximation to the MMSE estimate can be obtained by truncating the sum in (9.11).

Let \( \Omega \) be the set of indices \( i \) for which \( P(H_i|y) \) are significant, and then normalize \( P(H_i|y) \) for \( i \in \Omega \) so that they add up to one. Then we arrive at the following approximation to the MMSE estimate:

\[
\hat{h}_{\text{MMSE}} \approx \hat{h}_{\text{MMSE}} \triangleq \frac{1}{\sum_{j \in \Omega} P(H_j|y)} \sum_{i \in \Omega} P(H_i|y)E[h|y,H_i].
\] (9.12)

\(^4\)Here \( \delta(t) \) denotes the Dirac impulse.
By Bayes’ rule, we have
\[
\hat{h}_{\text{MMSE}} \triangleq \sum_{j \in \Omega} p(y | H_j) P(H_j) \sum_{i \in \Omega} P(H_i) p(y | H_i) E[h | y, H_i].
\] (9.13)

In order to evaluate (9.13), what remains is the following: (a) To compute \(p(y | H_i)\), (b) to compute \(E[h | y, H_i]\), and (c) to invent a mechanism for selecting \(\Omega\). Before we proceed with these tasks, let us introduce the following notation:

\[
\begin{align*}
x_j : & \text{ } j\text{th column of } X \\
\Gamma_i : & \text{ the set of indices } j \text{ for which } h_j \text{ are constrained to zero under } H_i \\
h_i : & \text{ } h \text{ with the elements corresponding to } \Gamma_i \text{ removed} \\
\bar{X}_i : & \text{ } X \text{ with the columns with indices in } \Gamma_i \text{ removed} \\
\tilde{h}_i : & \text{ } h \text{ with the elements not in } \Gamma_i \text{ removed (i.e., the opposite of } \bar{h}_i). \\
\end{align*}
\]

(Note: the length of \(\bar{h}_i\) equals \(n\) minus the number of elements in \(\Gamma_i\). The length of \(\tilde{h}_i\) is equal to the number of elements in \(\Gamma_i\).)

**Computation of \(p(y | H_i)\):** Conditioned on \(H_i\), we have that \(h_j = 0 \text{ for } j \in \Gamma_i\) and \(h_j \text{ i.i.d. } N(0, \gamma^2) \text{ for } j \not\in \Gamma_i\). Then, \(y | H_i \sim N(0, Q_i)\).

So,
\[
p(y | H_i) = \frac{1}{\sqrt{2\pi}^N |Q_i|^{1/2}} \exp \left( -\frac{1}{2} y^T Q_i^{-1} y \right).
\] (9.14)

Note that\(^5\)
\[
|Q_i| = \left| \gamma^2 \bar{X}_i \bar{X}_i^T + \sigma^2 I \right| = \sigma^{2N} \left| \frac{\gamma^2}{\sigma^2} \bar{X}_i \bar{X}_i^T + I \right| = \sigma^{2N} \gamma^2 \bar{X}_i^T \bar{X}_i + I
\] (9.15)

and\(^6\)
\[
Q_i^{-1} = \left( \gamma^2 \bar{X}_i \bar{X}_i^T + \sigma^2 I \right)^{-1} = \frac{1}{\sigma^2} I - \frac{1}{\sigma^2} \bar{X}_i \left( \frac{1}{\gamma^2} I + \frac{1}{\sigma^2} \bar{X}_i^T \bar{X}_i \right)^{-1} \bar{X}_i^T.
\] (9.16)

The expressions in (9.15) and (9.16) can be used to boost the computational efficiency: \(\bar{X}_i^T \bar{X}_i\) is much faster to evaluate than \(\bar{X}_i \bar{X}_i^T\) if \(N \gg n\). If \(n > N\) the original expressions should be used instead.

\(^5\)Using the identity \(|I + AB| = |I + BA|\).
\(^6\)Using the matrix inversion lemma.
9.3. MMSE Parameter Estimation

Computation of $E[h|y,H]$:

It is sufficient to find $E[\tilde{h}_i|y,H]$ and $E[\hat{h}_i|y,H]$, because $h$ is composed of $\tilde{h}, \hat{h}$. Clearly

$$E[\tilde{h}_i|y,H] = 0$$  \hspace{1cm} (9.17)

since the elements of $\tilde{h}_i$ are constrained to zero under $H_i$. We must now find $E[\hat{h}_i|y,H]$. Under $H_i$, $\hat{h}_i$ and $y$ are jointly Gaussian as follows

$$\begin{bmatrix} y \\ \hat{h}_i \end{bmatrix} | H_i \sim N \left( 0, \begin{bmatrix} Q_i & \gamma^2 \bar{X}_i \\ \gamma^2 \bar{X}_i^T & \gamma^2 I \end{bmatrix} \right).$$

Applying a standard result (see Theorem 10.2 of [75]), the conditional mean evaluates to

$$E[\hat{h}_i|y,H] = \gamma^2 \bar{X}_i^T Q_i^{-1} y.$$  \hspace{1cm} (9.18)

Selection of $\Omega$:

We now have the ingredients of the sum in (9.13); namely, equations (9.14), (9.17) and (9.18). What remains is to select the subset of $H_i$ over which the sum in (9.13) should be computed. That is, we must find the hypotheses $H_i$ which contribute significantly to (9.13). This is a hard task. We propose a strategy, based on successive model reduction, which results in the algorithm in the next subsection.

9.3.2 Algorithm for Computing the Approximate MMSE Estimate

1. Start with $i = 0$; i.e., consider $H_0$.

2. Compute the contribution of $H_i$ to (9.13).

3. Find out what coefficient $h_j$ would result in the largest $P(H_i|y)$, if $h_j$ were constrained to zero. That is, evaluate $P(H_k|y)$ for all $H_k$ which can be obtained from $H_i$ by constraining one more coefficient to zero, and then select the $H_k$ (i.e., let $i := j$), for which $P(H_k|y)$ is the largest. Note that for this task, it is sufficient to compute $p(y|H_k)P(H_k)$, instead of $P(H_k|y)$. Namely, by Bayes’ rule we have that $P(H_k|y) = p(y|H_k)P(H_k)/p(y)$ where $p(y)$ does not depend on $k$.

4. If $i = 2^n - 1$ (this must happen after precisely $n$ iterations), then compute the contribution of $H_i = H_{2^n - 1}$ to (9.13) and terminate. Otherwise, go to Step 2.

Note that the proposed algorithm will select only one hypothesis at each sparsity level, i.e., one hypothesis with no nonzero coefficients, one with precisely one nonzero coefficient, one with two nonzero coefficients, and so on. Naturally this leads to a somewhat disproportionate sampling of the hypothesis space, and consequently the algorithm is approximate (as already pointed out). However, this approximation appears to be an unavoidable consequence of the algorithm’s linear complexity in $n$ (see also the next subsection).
9.3.3 Computational Complexity

The computational complexity of our method is dominated by the selection of \( \Omega \). This selection is carried out \( n \) times (once for each value of \( i \) in Section 9.3.2). Each time, the matrix inverse \( Q^{-1} \) and the determinant \( |Q| \) must be computed \( \mathcal{O}(n) \) times (in order to find what coefficient should be eliminated next). The cost for computing \( Q^{-1} \) and \( |Q| \) is \( \mathcal{O}(Nn^2) \) if \( N > n \) and \( \mathcal{O}(N^2n) \) if \( n > N \). Consequently, the total computational cost for the method is dominated by a factor which behaves as \( \mathcal{O}(Nn^4) \) in the overdetermined case (when \( N > n \)), or \( \mathcal{O}(N^2n^3) \) in the underdetermined case (when \( n > N \)). This should be compared to the direct evaluation of (9.11) which would cost \( \mathcal{O}(2^nNn^2) \) when \( N > n \) or \( \mathcal{O}(2^nN^2n) \) if \( n > N \). That is, the approximation made when going from (9.11) to (9.12) reduces the complexity from being exponential in \( n \) to being polynomial in \( n \). This saving in computation can be very significant.

9.3.4 Implementation Considerations

In practice, \( p(y|H_i) \) can be a very small (or large) number. In a computer implementation, it is therefore better to work with \( \log(p(y|H_i)) \). Before the weighted sum in (9.13) is computed, one can then normalize \( \log(p(y|H_i)) \) by adding a constant \( \eta \) so that the “normalized” conditional pdf, \( e^{\eta} \cdot p(y|H_i) \), falls within the computer’s floating point operating range.\(^7\)

9.4 Model Selection

In practice, \( P(H_i|y) \) will never be exactly equal to zero. This means that the full model \( H_0 \) will always contribute to the sum in (9.12). Consequently, the algorithm in Section 9.3.2 will never result in a “perfectly sparse” estimate, even though many estimated coefficients often will be very close to zero.

By way of contrast, as discussed in Section 9.3, sometimes one is interested in the model selection problem, i.e., the task of finding the subset of regressors which best models the data. This amounts to choosing a model \( H_i \) among the \( 2^n \) possible models in (9.8). One can show that the selection rule which minimizes the probability of an incorrect choice (i.e., \( P(\hat{i} \neq i) \)) is to take

\[
\hat{i} = \text{argmax}_{i \in \{0, \ldots, 2^n - 1\}} P(H_i|y).
\]

(For a proof, see e.g., the discussion around (2.36) in Chapter 2.) Inspired by the developments in Section 9.3, we can devise a computationally feasible variant of (9.19) by restricting the search over \( i \) to those models encountered when running the algorithm in Section 9.3.2. This gives us

\[
\text{\(\hat{i}\)} = \text{argmax}_{i \in \Omega} P(H_i|y).
\]

Note that model selection is generally considered a difficult problem, and a host of techniques (derived under different paradigms) are available [150]. For\(^7\) A free implementation of our estimator (in C++) can be obtained from [85].
9.5 Strategies for Handling Unknown \( \{P(H_i)\}, \gamma^2, \sigma^2 \).

The model selection problem at hand, \( (9.20) \) along with the algorithm in Section 9.3.2 provides a systematic solution, which is optimal (provided \( \Omega \) contains the correct model) in the sense that under the assumptions made in Section 9.2, no other model selection method can have a higher probability of returning the correct model.

In this context, one may think of computing an approximation to the MMSE estimate by retaining only one term in \( (9.12) \), namely the one for which \( i = \hat{i} \) (i.e., the index \( i \) with the largest \( P(H_i|y) \) encountered in the model reduction). The resulting estimate of \( h \) is

\[
\hat{h}_{\text{one-term}} = E[h|y, H_{\hat{i}}], \quad \text{where } \hat{i} \text{ is given by } (9.20).
\]

Its computation essentially corresponds to “detection before estimation”, i.e., first trying to detect what coefficients are zero, and then estimate the rest via MMSE. We call the resulting estimate the “one-term” sparse MMSE estimate. It will turn out that in most cases the loss of retaining only one single term in \( (9.12) \) is relatively small, i.e., \( \hat{h}_{\text{one-term}} \) is close to \( \hat{h}_{\text{MMSE}} \).

9.5 Strategies for Handling Unknown \( \{P(H_i)\}, \gamma^2, \sigma^2 \)

In Sections 9.3 and 9.4 we have assumed that the parameters \( \{P(H_i)\}, \gamma^2, \sigma^2 \) are known. Given that one has correct knowledge of these parameters, the estimate we derived is optimal in the MMSE sense (up to the approximation in \( (9.12) \)); the model selection in \( (9.19) \) is optimal as well. However, in practice, one may not know \( \{P(H_i)\}, \gamma^2, \sigma^2 \) perfectly. One can then follow one of two possible strategies as outlined in the following two subsections. We restrict the discussion to the case that \( h_j \) are i.i.d. so that \( \{P(H_i)\} \) obey \( (9.9) \). The relevant parameters are then \( p, \sigma^2, \gamma^2 \).

9.5.1 Bayesian Estimator with known a Priori Distributions for the Parameters

The first approach is to assume that \( p, \gamma^2, \sigma^2 \) are independent random variables with known prior distributions, and average over these in the same way as we average over the model structures \( H_i \) in \( (9.11) \). More explicitly, this would give us

\[
\hat{h}_{\text{MMSE}} = E[h|y]
\]

\[
= \int_0^\infty \int_0^\infty \int_0^{1^2-1} \sum_{i=0}^{2^n} p(H_i, \sigma^2, \gamma^2, p|y) E[h|y, H_i, \sigma^2, \gamma^2, p] \, dp \, d\sigma^2 \, d\gamma^2
\]
and

\[ P(H_i|y) = \frac{p(y|H_i)P(H_i)}{p(y)} \quad (9.23) \]

\[ = \int_0^\infty \int_0^\infty \int_0^1 \frac{p(y|H_i, \sigma^2, \gamma^2, p)p(H_i, p)p(\gamma^2) dp \sigma^2 d\gamma^2}{p(y)} \]

where

\[ p(H_i, \sigma^2, \gamma^2, p|y) = \frac{p(y|H_i, \sigma^2, \gamma^2, p)p(H_i, p)p(\gamma^2) p(\sigma^2)}{p(y)}. \quad (9.24) \]

In the above equations, \( p(y) \) can be evaluated as follows:

\[ p(y) = \int_0^\infty \int_0^\infty \int_0^1 \sum_{i=0}^{2^n-1} p(y|H_i, \sigma^2, \gamma^2, p)p(H_i, p)p(\gamma^2) p(\sigma^2) dp \sigma^2 d\gamma^2. \quad (9.25) \]

If nothing at all is known about \( p, \sigma^2, \gamma^2 \), one may use non-informative densities for these (for example, a uniform distribution between 0 and 1 for \( p \), and a scaling invariant nonuniform prior for \( \sigma^2, \gamma^2 \); e.g., see Chapter 24 of [92], or possibly even an improper prior). If \( p, \gamma^2, \sigma^2 \) are known with some uncertainty, then the corresponding distributions describing this uncertainty can be used directly. In (9.22)–(9.25), \( p(y|H_i, \sigma^2, \gamma^2, p) \) and \( E[h|y, H_i, \sigma^2, \gamma^2, p] \) are the same as the corresponding expressions in Section 9.3. Unfortunately, evaluating the integrals in (9.22) is a computationally very burdensome problem. One may resort to numerical techniques such as Monte-Carlo methods [92], but in any event the complexity stays relatively high when compared to our suggested algorithm. An alternative to performing this integration is described in the next section.

### 9.5.2 Empirical Bayesian Version of the Estimator

The other avenue is to try to “estimate” the parameters \( p, \sigma^2, \gamma^2 \) from the data in some way. (This is also the paradigm often used to choose the parameter \( c \) for Lasso [157].) By doing so, one effectively approximates the \textit{a priori} distributions of \( p, \sigma^2, \gamma^2 \); e.g., see Chapter 24 of [92], or possibly even an improper prior). If \( p, \gamma^2, \sigma^2 \) are known with some uncertainty, then the corresponding distributions describing this uncertainty can be used directly. In (9.22)–(9.25), \( p(y|H_i, \sigma^2, \gamma^2, p) \) and \( E[h|y, H_i, \sigma^2, \gamma^2, p] \) are the same as the corresponding expressions in Section 9.3. Unfortunately, evaluating the integrals in (9.22) is a computationally very burdensome problem. One may resort to numerical techniques such as Monte-Carlo methods [92], but in any event the complexity stays relatively high when compared to our suggested algorithm. An alternative to performing this integration is described in the next section.

The other avenue is to try to “estimate” the parameters \( p, \sigma^2, \gamma^2 \) from the data in some way. (This is also the paradigm often used to choose the parameter \( c \) for Lasso [157].) By doing so, one effectively approximates the \textit{a priori} distributions of \( p, \sigma^2, \gamma^2 \); e.g., see Chapter 24 of [92], or possibly even an improper prior). If \( p, \gamma^2, \sigma^2 \) are known with some uncertainty, then the corresponding distributions describing this uncertainty can be used directly. In (9.22)–(9.25), \( p(y|H_i, \sigma^2, \gamma^2, p) \) and \( E[h|y, H_i, \sigma^2, \gamma^2, p] \) are the same as the corresponding expressions in Section 9.3. Unfortunately, evaluating the integrals in (9.22) is a computationally very burdensome problem. One may resort to numerical techniques such as Monte-Carlo methods [92], but in any event the complexity stays relatively high when compared to our suggested algorithm. An alternative to performing this integration is described in the next section.
For the problem at hand, we can devise the following estimators of the parameters $p, \sigma^2, \gamma^2$. Clearly, if $N > n$, an unbiased, consistent estimate of $\sigma^2$ can be easily obtained by taking

$$\hat{\sigma}^2 = \frac{1}{N-n} \|y - X \hat{h}_{LS}\|^2$$

(see (9.2)). Next, to estimate $p$ and $\gamma^2$, we note that under the model in Section 9.2 we have

$$E[h_j^2] = (1-p)\gamma^2$$

and

$$E[h_j^4] = 3(1-p)\gamma^4.$$

A simple moment-based estimator can then be obtained by taking

$$\hat{\gamma}^2 = \frac{\sum_{j=0}^{n-1} \hat{h}_{LS,j}^4}{3 \sum_{j=0}^{n-1} \hat{h}_{LS,j}^2}, \quad \hat{p} = 1 - \frac{3}{n} \left( \frac{\sum_{j=0}^{n-1} \hat{h}_{LS,j}^2}{\sum_{j=0}^{n-1} \hat{h}_{LS,j}^4} \right)^2$$

where $\hat{h}_{LS}$ is the unstructured LS estimate (see (9.2)). Note that one must make sure that $\hat{p} \geq 0$, for instance, by setting it to zero should the above expression happen to assume a negative value.

The estimates of $\sigma^2, \gamma^2, p$ proposed here are simple, intuitively appealing and statistically sound. However, they are not optimal in any sense. Naturally, we expect the estimates to be better the larger $n, N$ are. Also, we expect the estimators to break down as $p \to 1$, because in that case most of the coefficients $h_j$ will be zero and eventually there will be too few samples available to estimate $\gamma^2$.

### 9.6 Numerical Examples

We present several numerical examples where we evaluate the performance of our methods via Monte-Carlo simulation. In all cases, the regressor matrix $X$ was randomly chosen of dimension $150 \times 30$ (i.e., $N = 150$, $n = 30$). As performance measure we use the empirical MSE of the parameter estimates:

$$\text{empirical MSE} = \frac{1}{M} \sum_{m=1}^{M} \| \hat{h}^{(m)} - h^{(m)} \|^2$$

where $\hat{h}^{(m)}$ and $h^{(m)}$ denote the estimated and true parameter values for realization number $m$, and $M$ is the total number of Monte-Carlo runs. We took $M = 10000$.

We compare the new sparse MMSE estimator (9.12) and its one-term approximation (9.21) to the conventional LS estimate (9.2) and the non-sparse

---

8The problem of estimating $\sigma^2$ is ill-posed if $n \geq N$. If $n \geq N$, a possible approach is to regularize the problem by imposing a prior on $\sigma^2$. 

MMSE estimate (9.3). We consider the following three cases: (i) the estimators are supplied with the true values of $p$, $\gamma^2$ and $\sigma^2$, (ii) the case when these parameters are mismatched, and (iii) the case when they are estimated from the data as described in Section 9.5.2.

We also compare our algorithm with the Lasso method [157], using the official implementation of Lasso [33] (written in the R language [67]). The routines in [33] were used to automatically choose $c$. This way, Lasso does not exploit any knowledge of $\sigma^2$, $\gamma^2$, $p$ (indeed, it is not obvious how such knowledge could be exploited by Lasso). Therefore, Lasso should be compared to the empirical-Bayesian version (see Section 9.5.2) of our estimator. Note that our estimator (9.12) has the lowest MSE, by definition—and up to the approximation made when going from (9.11) to (9.12)—whenever the data are generated by the mechanism of Section 9.2. Thus, we can easily find scenarios where our method outperforms competitors like Lasso, and probably vice versa as well (by generating “appropriate” data). In this light, the rationale behind including the comparison with Lasso is to demonstrate to what level the performance can be increased by using appropriate estimators. Lasso was a natural choice as competing method, not only since it is widely used but also because there are free, numerically reliable implementations of it. Another possible comparison is against SBL [166]. In experiments not detailed here, we have observed that the performance of the method in [166] falls between that of Lasso and that of our empirical Bayesian estimator; we omit the corresponding plots to increase readability of the figures. Furthermore, since Lasso generally gives an estimate of $h$ for which some elements are exactly equal to zero, it can be used as a model (or “basis”) selection tool and thus it is a meaningful baseline for the corresponding comparisons as well (see Example 4).

9.6.1 Example 1: Estimation Performance Comparison with Uncorrelated Regressors

In this example we used a regressor matrix $X$ with i.i.d. $\mathcal{N}(0, 1)$ elements. This regressor matrix is well-conditioned in general. (Typically, the singular value spread of $X$ was about 3 dB.) As for the parameter vector, we generated $n$ independent coefficients $h_j$ for each realization of $h$. The coefficients $h_j$ were distributed according to $\mathcal{N}(0, \gamma^2_0)$ but set to zero with probability $p_0$. In this example we tested two different values of $p_0$. The noise variance, $\sigma^2_0$, was varied from 5 dB down to $-20$ dB.

We first tested the methods on sparse data, setting $\gamma_0 = 1$ and $p_0 = 0.5$ (i.e., 50% sparsity). Figure 9.1 shows the results. We show results for (i) our sparse MMSE estimator with parameters perfectly matched to the data model (i.e., we used $p = p_0 = 0.5$, $\gamma^2 = \gamma^2_0 = 1$, $\sigma^2 = \sigma^2_0$ in the estimator); (ii) the empirical Bayesian variant of the estimator (which estimates $p$, $\gamma^2$, $\sigma^2$ from the data); (iii) Lasso; (iv) the non-sparse MMSE estimate (9.3) (supplied with $\sigma^2_0$ and $\gamma^2_0$); and (v) the LS estimate (9.2). Note that both variants of our estimator outperform Lasso. This is no surprise as the model underlying our

\footnote{In this section we shall use the subscript $(\cdot)_0$ to denote the “true” parameter values, i.e., those used when generating the data.}
estimate is matched (up to the approximation in (9.12)) to the data generating mechanism. Furthermore, our empirical Bayesian estimator shows almost as good performance as our Bayesian estimator which exploits all available \textit{a priori} knowledge.

For Lasso, we also tried providing it with the “true” value of $c$ for each realization, that is, setting $c = \sum_j |h_j|$, where $\{h_j\}$ denotes the actual true values of the coefficients of the parameter vector. (This was the value of $c$ that appeared to give the best result for Lasso.) However, even with this very strong, realization specific \textit{a priori} knowledge, our method outperformed Lasso. We omit the corresponding plots for brevity.

The loss in MSE performance associated with the retention of only one of $n$ terms in (9.12) is small, as can be seen when comparing the performance of (9.12) and (9.21) in Figure 9.1.\footnote{This observation holds for all examples. However, to increase readability of the figures, we exclude (9.21) from the remaining figures.} This indicates that the approximation made when going from (9.11) to (9.12) probably is small, and thus the MSE of (9.12) practically also serves as a lower bound on the achievable estimation performance.

Next we examine what happens if the estimator is mismatched to the data. More precisely, we consider using the sparse MMSE estimator (with $p = 0.5$) when the data are actually not sparse (i.e., $p_0 = 0$). We use the same assumptions as in Figure 9.1 regarding $\sigma^2, \gamma^2$. In Figure 9.2, we illustrate the performance of the estimators. Not surprisingly, the sparse estimator using $p = 0.5$ is somewhat worse than the non-sparse MMSE estimate (9.3), which is optimal in this scenario. However, the gain in Figure 9.1 is larger than the loss in Figure 9.2. Neither Lasso nor the empirical Bayesian variant of our estimator shows a significant loss compared to (9.3).

\subsection*{9.6.2 Example 2: Analysis of the Sensitivity to the \textit{a Priori} Parameters}

Figures 9.3, 9.4 and 9.5 shed additional light on the issue of mismatch between the estimator and the “reality” (i.e., the parameters used when generating the testing data).

Figure 9.3 examines the robustness of the estimator when $p \neq p_0$. Here we generated data with different sparseness by varying $p_0$. The other parameters were set to $\sigma^2 = \sigma_0^2 = -15$ dB and $\gamma^2 = \gamma_0^2 = 1$. We show the performance of the conventional MMSE, the sparse MMSE with $p = 0.3, 0.5$ and 0.8, and the Lasso. The results show that the proposed estimate is robust in the sense that it improves performance much more in a scenario with a sparse parameter vector, than what it reduces the performance in a non-sparse (mismatched) scenario. This is a very desirable property for an estimator. Indeed, the sparse MMSE estimate outperforms the non-sparse MMSE estimate except for when $p_0 < 0.1$ (cf. also Figure 9.2). Note that the MMSE estimate with the correct value of $p = p_0$ always gives the best performance, as one would expect. Also, note that the sparse MMSE estimates outperform Lasso, even when $p$ is severely
mismatched. We further note a performance drop for our empirical Bayesian estimator as $p_0 \to 1$. This is natural, as pointed out at the end of Section 9.5.2.

Figure 9.4 examines how sensitive the estimator is when $\gamma^2 \neq \gamma_0^2$ (the other parameters were set to $\sigma^2 = \sigma_0^2 = -15$ dB and $p = p_0 = 0.5$). The conclusion from this figure is similar to that of Figure 9.3: the estimator is fairly insensitive to using the correct $\gamma^2$.

Figure 9.5 examines what happens if $\sigma^2 \neq \sigma_0^2$ (with the other parameters set to $\gamma^2 = \gamma_0^2 = 1$ and $p = p_0 = 0.5$). Also here, the performance is relatively insensitive to a parameter mismatch. However, using a value of $\sigma^2$ which significantly exceeds the true noise variance is harmful to the performance. Fortunately, however, $\sigma^2$ is rather easy to estimate from the data (at least if $N > n$); see Section 9.5.2.

To summarize, we have found that the estimator is very robust to mismatches between the true values of the a priori parameters, and those supplied to the estimator. This fact may be one contributing reason to why the empirical Bayesian estimator (see Section 9.5.2) works so well, despite using suboptimal estimates of $\sigma^2$, $\gamma^2$ and $p$.

9.6.3 Example 3: Estimation Performance Comparison with Correlated Regressors

This experiment is identical to Example 1, except for that here we use a regressor matrix $X$ with correlated columns. More precisely, we let $X$ be a Toeplitz matrix whose columns are shifted realizations of zero-mean, white Gaussian noise filtered through an all-pole (autoregressive) filter with poles in $z = 0.99 e^{\pm i\pi/4}$. (Typically, the singular value spread of $X$ was about 15 dB.) This example corresponds to estimation of finite impulse response (FIR) filter coefficients [147]. Like in Example 1, we took $\gamma^2 = \gamma_0^2 = 1$, $p = p_0 = 0.5$ and varied $\sigma^2 = \sigma_0^2$. Figure 9.6 shows the results.

The gain of using a sparse estimate here is even larger than in Example 1. This is so because in the present example the elements of $y$ are strongly correlated. Consequently, each element of $y$ contains much less information than in Example 1, and the influence of the a priori assumptions becomes stronger. Also in this example, our estimator outperforms Lasso.

We have also performed simulations for the case $p_0 = 0$ where our estimator was used with $p = 0.5$ (corresponding to Figure 9.2). The same conclusions as in Figure 9.2 hold here, but for brevity, we omit the plot.

9.6.4 Example 4: Model Selection

Here we illustrate the model selection algorithm of Section 9.4. The model $H^i$ with $\hat{i}$ given by (9.20) is taken as the “detected” model. The simulation setup is the same as in Figure 9.1. Figure 9.7 shows the percentage $\rho$ of regressors (or equivalently, of the coefficient values $h_j$) which are correctly identified as being active and inactive, respectively. As expected, the empirical version of the estimator (which does not need any information on $\gamma^2, \sigma^2, p$) performs
9.7. Concluding Remarks

somewhat worse than the estimator provided with the true values of these parameters.

Our model estimate $H^\hat{i}$ uniformly outperforms Lasso. Indeed, the fraction of correctly identified structures, say $\rho_s$, behaves as $\rho_s \approx \rho^n$. A more accurate look showed that $\rho_s = 50\%$ already at $\sigma^2_0 \approx -10$ dB, whereas at the same signal-to-noise-ratio, Lasso correctly identifies only a few percent of the models. This does not come as a total surprise, as it is known that Lasso, when the threshold $c$ is chosen to minimize the prediction error (this is the case for the official implementation we used), exhibits weak performance on model selection, even asymptotically (see, e.g., [88] or results in Examples 3 and 4 of [157]; also for Lasso’s close relative Lars see the comments on page 456 of [34]). Better performance for Lasso could likely be obtained by using another method for choosing $c$.

9.7 Concluding Remarks

We have considered linear regression with a sparse coefficient vector, using an explicit model for the parameter vector. The model has a strong intuitive appeal and it can be parameterized by three parameters, all of which have clear physical interpretations.

Depending on the application, one may be primarily interested in either (i) to obtain the best possible parameter estimate, or (ii) to select a reduced model (i.e., one with only a few regressors). Our framework simultaneously delivers a solution to both problems (i) and (ii). Namely, the answer to (i) is the estimate in (9.12) and the solution to (ii) is the model index in (9.20). We also showed that inference based on our model is very robust to mismatches in the a priori knowledge about the model. Encouraged by this observation, we derived an empirical-Bayes version of the estimate, which estimates the a priori parameters from the data, thus obviating the need for selecting any user parameters. We showed that both our standard Bayesian estimator and its empirical-Bayes version have very good performance, both in terms of MSE of the parameter estimate and when used for model selection.
Figure 9.1: Performance example for a system with $n = 30$ coefficients and $N = 150$ observations. The regressor matrix ($X$) had i.i.d. $\mathcal{N}(0,1)$ elements. The coefficient vector had i.i.d. $\mathcal{N}(0,1)$ elements and each element was set to zero with probability $p_0 = 0.5$. The sparse MMSE estimator was provided with $\sigma^2 = \sigma_0^2$, $\gamma^2 = \gamma_0^2 = 1$ and $p = p_0 = 0.5$ (by way of contrast, the empirical-Bayesian version of our algorithm estimated these parameters from the data). The closeup shows the region $\sigma_0^2 \in [-10, -12.5]$ in greater detail.
Figure 9.2: Same as Figure 9.1, but here the coefficient vector had i.i.d. \( N(0, 1) \) elements (i.e., \( p_0 = 0 \)). The sparse MMSE estimator was provided with \( \sigma^2 = \sigma_0^2, \gamma^2 = \gamma_0^2 = 1 \) but with a mismatched value of \( p_0 \). \( p = 0.5 \neq p_0 \) (by way of contrast, the empirical-Bayesian version of our algorithm estimated these parameters from the data).
Figure 9.3: Performance example: $p$-mismatch for $\gamma^2 = \gamma_0^2 = 1$, $\sigma^2 = \sigma_0^2 = -15$ dB.
Figure 9.4: Performance example: $\gamma^2$-mismatch for $\sigma^2 = \sigma_0^2 = -15$ dB, $p = p_0 = 0.5$. 
Figure 9.5: Performance example: $\sigma^2$-mismatch for $\gamma^2 = \gamma_0^2 = 1$, $p = p_0 = 0.5$. 
Figure 9.6: Same as Figure 9.1, but here the regressor matrix ($X$) has correlated columns (formed from an autoregressive random process).
Figure 9.7: Illustration of how the method can be used for model selection. All parameters are the same as in Figure 9.1.
Chapter 10

RAKE Receiver for Channels with a Sparse Impulse Response

We derive the optimal receiver for RAKE diversity combining [114] on channels with a sparse impulse response. The receiver is based on the Bayesian philosophy and thus it requires the knowledge of certain \textit{a priori} parameters. However, we also derive an empirical Bayesian version of our receiver, which does not require the choice of the \textit{a priori} parameters. We show that both versions of our detector can outperform a classical training-based maximum-ratio-combining detector.

10.1 Introduction

Diversity reception is a key component in many communication systems. To optimally exploit it, one must implement the receiver that determines the most likely transmitted message given the received data. This problem is complicated because normally the receiver has only imperfect (or partial) knowledge of the channel. In this context, a substantial body of work [7; 9; 29; 40; 49; 51; 113; 136; 138; 159] quantifies the performance loss when using an estimated (or for other reasons imperfect) channel in lieu of the true one for symbol detection. Less work is available on the design of detectors that can optimally make use of an estimated channel. However, optimal decision metrics are known for some cases; see, e.g., [42] for single-channel (i.e., no diversity) reception and [155] for space-time codes.

In this chapter we are concerned with diversity combining for \textit{sparse} channels, i.e., channels where the gains for many of the diversity paths are zero.
The main motivation for this is RAKE combining for spread-spectrum systems [114] operating on channels where the signal bandwidth is so large that individual multipath components can be resolved, so that the chip-sampled impulse response is sparse. Basically, there are two types of RAKE receivers in the literature: either the RAKE fingers are placed at fixed locations (typically on a grid), or their locations are adaptively chosen via a finger search mechanism [25]. Fixed-grid RAKE usually performs well if there are enough many RAKE fingers [26]. Also, its corresponding channel model may be sparse. RAKE receivers with path searching, however, do generally not give sparse channel models. (If the finger search algorithm succeeds, it positions the fingers at the correlation lags where the impulse response has power. The resulting effective channel will then be non-sparse.) Therefore fixed-finger RAKE is the main motivation for our work. In this context the main goal of our work is to examine how much performance one can gain by knowing that the channel is sparse. By way of contrast, if such knowledge were available but not exploited, some of the channel taps included in the detection process would carry only noise (no signal). This would lead to unnecessary noise enhancement and reduced detection performance.

Receivers for sparse channels have previously been studied in some special cases. For example, [115] derived a generalized Akaike information criterion to estimate the channel structure for OFDM systems. However, the optimal diversity combiner for sparse channels, derived in this chapter, is novel to our knowledge.

10.2 Problem Formulation

We consider a setup where the transmitter first sends a training (pilot) sequence \( s_t(k), k = 0, \ldots, n_t - 1 \), which is known to the receiver. Thereafter, the transmitter sends an information symbol \( s \), to be detected by the receiver. For simplicity of the exposition we restrict the discussion to M-PSK modulation (so, \(|s| = |s_t(k)| = 1, k = 0, \ldots, n_t - 1 \)). We assume that the transmitted data are spread with an ideal spreading sequence, and then matched-filtered and despread at the receiver. The data propagate over a frequency selective channel with the chip-sampled impulse response \( h = [h_0, \ldots, h_{n_t - 1}]^T \). We further assume that the received data are disturbed by circular AWGN, but that there is no multiaccess interference. The received data can be written as

\[
\begin{align*}
    y_t(0) & = h s_t(0) + e_t(0), & y_t(0) \in \mathbb{C}^n \\
  \vdots \\
    y_t(n_t - 1) & = h s_t(n_t - 1) + e_t(n_t - 1), & y_t(n_t - 1) \in \mathbb{C}^n \\
    y & = h s + e, & y \in \mathbb{C}^n
\end{align*}
\]

where the noise terms \( e_t(k) \) and \( e \) are i.i.d. \( \mathcal{CN}(0, \sigma^2 I) \). (Throughout, the subscript \((\cdot)_t\) denotes quantities associated with the training.) The objective for the receiver is to detect \( s \) given \( \{y_t(k)\}, \{s_t(k)\} \) and \( y \).

Naturally, if \( h \) were known to the receiver the problem would be trivial, because \( h^H y / \|h\|^2 \) is a sufficient statistic for the detection of \( s \). The receiver
10.3. The Optimal Receiver

would then take (assuming \( P(s) \) is uniform for all \( s \)),

\[
\hat{s} = \arg \min_s \left| s - \frac{h^H y}{\|h\|^2} \right|^2.
\]

If \( h \) is unknown, however, the optimal receiver will in general depend on the specific assumptions made and on the philosophy used for the receiver design. Most commonly, an \textit{a priori} density is assumed for \( h \), and the optimal receiver is derived under that assumption. However, it is also possible to take an “orthodox” approach (not making any assumptions on \( h \)), for example using a generalized-likelihood ratio test. In this chapter (like in most classical communication theory literature) we stick to the Bayesian paradigm, since it enables unique, well-defined inference. Additionally, it makes all assumptions explicit.

We shall assume for the purpose of receiver design that, with some probability, some elements of \( h \) are equal to zero. More precisely, we model \( \{ h_i \} \) as i.i.d. random variables with the following mixture distribution:

\[
h_i = \begin{cases} 
0, & T_i = 0 \\
\mathcal{C}N(0, \rho_i^2), & T_i = 1
\end{cases}
\]

where \( \{ T_i \}_{i=0}^{n-1} \) are i.i.d. Bernoulli random variables with parameter \( p_i \): \( T_i \in \{0,1\} \) and \( P(T_i = 0) = p_i = 1 - P(T_i = 1) \). This is not the only possible model, but it does seem to make sense from a physical perspective (for example, \( |h_i| \) will have a Rayleigh distribution when \( T_i = 1 \)), and also it leads to tractable mathematics.

10.3 The Optimal Receiver

The optimal receiver finds the \( s \) which has the largest probability of being transmitted, given all received data. Let us introduce the notation \( Y = [y_t(0) \cdots y_t(n_t - 1) y] \in \mathbb{C}^{n_t \times (n_t + 1)}. \) Then the receiver should find \( \hat{s} = \arg \max_s P(s|Y) \). (In what follows, the training sequence \( \{ s_t(k) \} \) is assumed to be known to the receiver.) To detect individual bits \( b_k \) in \( s \), we should instead maximize \( P(b_k = \beta|Y) = \sum_{s: b_k=\beta} P(s|Y). \) To take soft decisions on a bit \( b_k \), the quantity of interest would be \( L_k = \log(P(b_k = 1|Y)/P(b_k = 0|Y)). \) Either way, what is important is to evaluate \( P(s|Y) \).

For equiprobable symbols (an assumption that can be relaxed) we have that

\[
P(s|Y) \cong p(Y|s) = \prod_{i=0}^{n-1} p(y_i|s)
= \prod_{i=0}^{n-1} \left( p_i p(y_i|s, T_i = 0) + (1 - p_i) p(y_i|s, T_i = 1) \right)
\]

where \( y_i = [y_t(0) \cdots y_t(n_t - 1) y] \), i.e., all data received for tap \( i \). Note that (10.2) only holds if the tap gains \( \{ h_i \} \) are independent and the noise samples

\[1\] Generally, we let subscript \((\cdot)_i\) denote the \( i \)th element of a vector.

\[2\] Hereafter “\( \cong \)” stands for equality up to irrelevant constants.
(elements of \{e_t(k)\} and \(e\)) are independent, so that \(\{y_i\}\) are independent (conditioned on \(s\)). This was, however, an assumption that we did make. To compute (10.2) we need the conditional densities \(p(y_i|s, T_i)\). For \(T_i = 0\) this follows directly, since in this case \(y_i \sim \mathcal{CN}(0, \sigma^2 I)\) and

\[
p(y_i|s, T_i = 0) = \frac{1}{\pi^{n_t+1}} \frac{1}{\sigma^{2(n_t+1)}} \exp\left(-\frac{1}{\sigma^2} \left( |y_i|^2 + \sum_{k=0}^{n_t-1} |e_t(k)|^2 \right) \right).
\]

Next, observe that, conditioned on \(s\) and \(T_i = 1\), we have \(y_i \sim \mathcal{CN}(0, Q_{s,i})\) where

\[Q_{s,i} \triangleq \rho_i^2 s s^H + \sigma^2 I\]

and \(s = [s_t(0) \cdots s_t(n_t - 1)]^T\) (\(Q_{s,i}\) is a function of \(s\)). So,

\[
p(y_i|s, T_i = 1) = \frac{1}{\pi^{n_t+1}} \frac{1}{|Q_{s,i}|} \exp\left(-y_i^H Q_{s,i}^{-1} y_i \right).
\]

Equation (10.4) can be simplified as follows. By the matrix inversion lemma,

\[
Q_{s,i}^{-1} = \frac{1}{\sigma^2} I - \frac{\rho_i^2}{\sigma^4 + \rho_i^2 \sigma^2 (n_t + 1)} s s^H,
\]

so

\[
y_i^H Q_{s,i}^{-1} y_i = \frac{1}{\sigma^2} \left( \sum_{k=0}^{n_t-1} |y_t(k)|^2 + |y_i|^2 \right) - \frac{\rho_i^2}{\sigma^4 + \rho_i^2 \sigma^2 (n_t + 1)} \left( \sum_{k=0}^{n_t-1} y_t^*(k) s_t(k) + y_i^* s \right)^2.
\]

Next, by using the identity \(|I + AB| = |I + BA|\) we have

\[|Q_{s,i}| = \sigma^{2(n_t+1)} \left( 1 + (n_t + 1) \frac{\rho_i^2}{\sigma^2} \right) \]

To summarize, we have found that the log-densities of the data are

\[
\log p(y_i|s, T_i = 0) \equiv -(n_t + 1) \log(\sigma^2) - \frac{1}{\sigma^2} \left( \sum_{k=0}^{n_t-1} |y_t(k)|^2 + |y_i|^2 \right)
\]

and

\[
\log p(y_i|s, T_i = 1) \equiv -(n_t + 1) \log(\sigma^2) - \frac{1}{\sigma^2} \left( \sum_{k=0}^{n_t-1} |y_t(k)|^2 + |y_i|^2 \right) - \log \left( 1 + (n_t + 1) \frac{\rho_i^2}{\sigma^2} \right) + \frac{\rho_i^2}{\sigma^4 + \rho_i^2 \sigma^2 (n_t + 1)} \left( \sum_{k=0}^{n_t-1} y_t^*(k) s_t(k) + y_i^* s \right)^2.
\]
10.3. The Optimal Receiver

Assembling things together and letting $J(a, b)$ stand for the Jacobian logarithm, $J(a, b) \triangleq \log(e^a + e^b)$, we have (omitting some irrelevant terms)

$$
\log P(s|Y) \equiv \sum_{i=0}^{n-1} \log \left[ p_i P(y_i|s, T_i = 0) \right. \\
\left. + (1 - p_i) P(y_i|s, T_i = 1) \right] \cong \sum_{i=0}^{n-1} J(\alpha_i, \beta_i)
$$

where $\alpha_i = \log(p_i)$ and

$$
\beta_i = \log(1 - p_i) - \log \left( 1 + (n_t + 1) \frac{\rho_i^2}{\sigma^2} \right) \\
+ \frac{\rho_i^2}{\sigma^4 + \rho_i^2 \sigma^2 (n_t + 1)} \left| \sum_{k=0}^{n_t-1} y_i^*(k)s_t(k) + y_i^* s \right|^2
$$

(10.5)

(Note that the entire right hand side of (10.3) is independent of $s$ and the same expression also appears in $p(y_i|s, T_i = 1)$. Therefore it cancels in the above equation.)

In the limit when all $p_i$ approach zero (i.e., there is no sparseness in the model), then the optimal receiver (10.5) becomes

$$
\log P_{\text{non-sparse}}(s|Y) \cong \sum_{i=0}^{n-1} \left[ \frac{\rho_i^2}{\sigma^4 + \rho_i^2 \sigma^2 (n_t + 1)} \left| \sum_{k=0}^{n_t-1} y_i^*(k)s_t(k) + y_i^* s \right|^2 \right].
$$

(10.6)

If $\rho_i^2 = \rho^2, \ i = 0, \ldots, n - 1$, the receiver in (10.6) is equivalent to MRC (maximum-ratio combining) using the LS/ML (least squares/maximum likelihood) channel estimate, as far as hard decisions are concerned. To see this, consider the LS/ML estimate of $h$:

$$
\hat{h}_{\text{LS}} = \frac{1}{n_t} \sum_{k=0}^{n_t-1} y_t(k)s_t(k).
$$

(10.7)

The MRC receiver using this estimated channel minimizes, with respect to $s$,

$$
\frac{s - \hat{h}_{\text{LS}}^H y}{||\hat{h}_{\text{LS}}||^2} = \frac{\left|s\right|^2}{||\hat{h}_{\text{LS}}||^2} + \frac{\hat{h}_{\text{LS}}^H y}{||\hat{h}_{\text{LS}}||^2} - 2 \Re \left( \frac{\hat{h}_{\text{LS}}^H y}{||\hat{h}_{\text{LS}}||^2} s^* \right) \\
\cong -\Re \left( \frac{\hat{h}_{\text{LS}}^H y s^*}{||\hat{h}_{\text{LS}}||^2} \right).
$$

(10.8)

This can be implemented with good numerical stability and high computational efficiency.

For example, one can show that $J(a, b) = \max(a, b) + \log(1 + e^{-|a-b|})$. The last term can be implemented via a table-lookup as a function of $a - b$. 
The receiver (10.6) maximizes, with respect to \( s \) (when \( \rho_i^2 = \rho^2 \), \( i = 0, \ldots, n - 1 \)),

\[
\sum_{i=0}^{n-1} \sum_{k=0}^{n_i-1} y_n^i(k)s_t(k) + y_n^i = \| \hat{h}_{ls} n_t + y s^* \|^2 \\
= n_i^2 \| \hat{h}_{ls} \|^2 + |s|^2 \| y \|^2 + 2 \text{Re} \left( n_i \hat{h}_{ls}^H y s^* \right) \\
\equiv \text{Re} \left( \hat{h}_{ls}^H y s^* \right) .
\]  

Comparison of (10.8) with (10.9) proves the equivalence. The same conclusion holds if the receiver instead performs MMSE (minimum mean square error) channel estimation followed by MRC, but also only under the assumptions that \( \rho_i^2 = \rho^2 \) and \( |s| = 1 \).

Finally, note that the channel often is constant over many symbols and one may then want to detect several symbols using the same channel estimate. It is straightforward to extend the above optimal receivers to detect a vector of information symbols. However, the complexity of the resulting detector will be exponential in the number of information symbols. A more pragmatic approach is to maximize (10.5) with respect to each symbol separately. Strictly speaking, the so-obtained receiver is suboptimal, but its complexity is linear in the number of symbols.

### 10.4 An Empirical Bayesian Receiver

Our receiver is Bayesian and it requires knowledge of \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1}, \sigma^2 \). How can we use this receiver if these parameters are unknown? There are several possible answers to this question. One could treat \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1}, \sigma^2 \) as random variables with certain \textit{a priori} densities and then average \( P(s|Y) \) over these. Alternatively, one could maximize (concentrate) \( P(s|Y) \) with respect to \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1}, \sigma^2 \), treating them as hyperparameters. However, we choose another approach; namely to estimate \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1}, \sigma^2 \) from the data.

Estimating \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1}, \sigma^2 \) from the data is a pragmatic (but in a strict sense suboptimal) solution and the resulting receiver will be a so-called “empirical Bayesian” detector [50]. We will restrict the discussion to i.i.d. \{\( h_i \)\}, so that \( \rho_i^2 = \rho^2 \) and \( p_i = p \) for \( i = 0, \ldots, n - 1 \). This is not the most general approach, but it facilitates the use of simple moment-based estimators for \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1} \). Also, our Bayesian receiver turns out to be robust in the sense that it generally shows good performance even with mismatched \( \{p_i\}_{i=0}^{n-1}, \{\rho_i^2\}_{i=0}^{n-1}, \sigma^2 \) (see Section 10.5). Thus, accurate estimation of these parameters does not seem crucial.

**Estimation of \( \sigma^2 \)**: An unbiased\(^4\), consistent estimate of \( \sigma^2 \) can be obtained

\(^4\)Note that we normalize with \( n_{1m} - n \) instead of just \( n_{1m} \) to take into account the fact that the same pilot data are used to compute \( \hat{h}_{ls} \) and \( \hat{\sigma}^2 \).
by taking
\[ \hat{\sigma}^2 = \frac{1}{n_t} \sum_{k=0}^{n_t-1} \| y_t(k) - \hat{h}_{LS}s_t(k) \|^2 \]
where \( \hat{h}_{LS} \) is obtained from (10.7).

**Estimation of \( \rho^2, p \):** Under the model (10.1) we have \( E[|h_i|^2] = (1 - p)\rho^2 \) and \( E[|h_i|^4] = 2(1 - p)\rho^5 \). By combining these equations and using \( E[|h_i|^2] \approx \sum_{i=0}^{n_t-1} |h_{LS,i}|^2/n \), \( E[|h_i|^4] \approx \sum_{i=0}^{n_t-1} |h_{LS,i}|^4/n \), we obtain the following estimates:
\[ \hat{\rho}^2 = \frac{\sum_{i=0}^{n_t-1} |h_{LS,i}|^4}{2 \sum_{i=0}^{n_t-1} |h_{LS,i}|^2} \quad \text{and} \quad \hat{p} = 1 - \frac{2}{n} \left( \frac{\sum_{i=0}^{n_t-1} |h_{LS,i}|^2}{\sum_{i=0}^{n_t-1} |h_{LS,i}|^4} \right)^2. \]

One must make sure that \( \hat{p} \geq 0 \), for instance, by setting it to a small positive value (or zero) if the above estimate turns out negative.

The estimates of \( \sigma^2, \rho^2, p \) proposed here are intuitively appealing and statistically sound, but not “optimal”. Naturally, we expect the estimates to improve with increasing \( n_t \). Also we expect that they will break down when \( p \to 1 \), and that they will not work very well for small \( n_t \). In particular, they will break down completely for \( n_t = 1 \), since then \( \hat{\sigma}^2 \) is undefined.

### 10.5 Numerical Examples

We illustrate the performance of our receivers via Monte Carlo simulation. We consider both receivers whose \textit{a priori} parameters are matched to the channel, and mismatched receivers. Also, we consider both a simple “toy” channel and a more realistic channel model for indoor radio [123]. In all examples we use QPSK modulation and show the bit-error-rates (BERs) for (i) the coherent receiver, (ii) the non-sparse Bayesian receiver (10.6) (same as MRC using the LS/MMSE estimated non-sparse channel), (iii) our sparse Bayesian receiver (10.5) and (iv) its empirical version.

#### 10.5.1 Sparse Toy Channel Model

We first consider a sparse toy channel where the \textit{a priori} parameters of the receiver are matched to the channel. This means that the channel impulse response is generated with the same parameters values as used in the receiver. The parameters are set to \( p_i = 0.9^5, \rho_i^2 = 1, n = 20 \), and \( h_i \) are i.i.d. We consider two choices of the training sequence length: \( n_t = 2 \), and \( n_t = 10 \). The receiver performances are shown in Figure 10.1. Our sparse Bayesian receiver outperforms the non-sparse Bayesian receiver with about 1.5 dB for both choices of \( n_t \). This is expected, as the non-sparse Bayesian receiver uses mismatched channel knowledge (\( p_i = 0 \) instead of \( p_i = 0.9 \)). The sparse Bayesian

---

5Using, e.g., that, when \( x \sim \mathcal{CN}(0, \lambda^2) \), \( E[x_1x_2x_3x_4] = E[x_1x_2]E[x_3x_4] + E[x_1x_3]E[x_2x_4] + E[x_1x_4]E[x_2x_3] \) [68].

6To avoid an error floor, we make sure that the channels \( h \) have at least one non-zero tap (when all taps are zero, no signal reaches the receiver and detection will be pure guesswork).
Figure 10.1: Toy channel model, 20 i.i.d. taps. Each tap is $\mathcal{CN}(0,1)$ with probability 10%, and zero with probability 90%. The receiver parameters $(p_i, \rho_i^2, \sigma^2)$ were the same as for the data generating process, i.e., $\rho_i^2 = 1, p_i = 0.9$.

receiver with the long training sequence performs about 1 dB worse than the coherent receiver. The performance of our empirical Bayesian receiver beats that of the non-sparse Bayesian receiver by about 1 dB, even with only two training samples.

10.5.2 Non-Sparse Toy Channel Model—Robustness to Mismatch

Next, we consider a non-sparse toy channel example where the data are generated as above, but with $n_t = 10, n = 10$ and $p_i = 0$ (i.e., all taps are non-zero). We use a mismatched sparse Bayesian receiver with $p_i = 0.5$. The BERs are shown in Figure 10.2. It is no surprise that the non-sparse Bayesian receiver performs slightly better than our (mismatched) sparse Bayesian receiver, as the former uses the true $p_i = 0$. However, the penalty for assuming sparseness is very small. Our empirical Bayesian estimator shows impressive performance—almost identical to that of the non-sparse Bayesian receiver, which is optimal in this scenario.
Figure 10.2: Toy channel model, 10 i.i.d. taps. Here all taps are \( \mathcal{C} \mathcal{N}(0, 1) \), i.e., there is no sparsity in the channel impulse response. The “Bayesian sparse” receiver uses the true values of \( \sigma^2 \) and \( \rho_i^2 \), but a mismatched value of \( p_i \) (it takes \( p_i = 0.5 \)). A training sequence of length \( n_t = 10 \) was used. The closeup shows the region \( \sigma^2 \in [0, -2.5] \) dB.
10.5.3 Indoor Radio Channel Model

Finally, we consider the indoor radio channel model of Saleh-Valenzuela [123]. This model is based on an extensive set of measurements in the 1.5 GHz band. We use the channel model with the parameter values suggested in [123] \((1/\Lambda = 300 \text{ ns}, 1/\lambda = 5 \text{ ns}, \Gamma = 60 \text{ ns}, \gamma = 20 \text{ ns})\). We assume a sampling interval of \(T_s = 10 \text{ ns}\), corresponding to a 100 MHz signal bandwidth, and root-raised-cosine filtering with a rolloff factor of \(\beta = 0.5\) at the transmitter and at the receiver. We further assume that the receiver can acquire approximate timing, so that the first sample in the impulse responses is taken uniformly at random between \(-1.5T_s\) and \(-0.5T_s\) relative to the timing of the first multipath ray. We truncate the channel impulse responses after \(n = 40\) taps and we use \(n_t = 5\) training symbols.

One may ask whether, or how well, the channel model assumed in Section 10.2 corresponds to reality, or how well it matches a more realistic or specific model like that of [123]. Specifically, one might ask whether the distribution we assumed for the channel coefficients \(\{h_i\}\) is reasonable. Under the model of [123] we can argue as follows. Owing to the random sampling timing, the pulse shaping, and since rays generally arrive in clusters according to [123] we cannot expect the independence assumption we made on \(\{h_i\}\) to hold exactly. However, \(\{h_i\}\) will be Rayleigh fading for the model of [123], because all channel taps in continuous time are complex Gaussian and \(\{h_i\}\) are obtained via a linear operation on these. This is demonstrated in Figure 10.3, where we show “empirical pdfs” of the magnitudes of a few taps based on 10000 Saleh-Valenzuela channel realizations. This figure also illustrates how the average tap power varies as a function of tap delay.

It is interesting to study the extreme cases when the receiver is supplied much a priori information, and no such information, respectively. To simulate a scenario with relatively much a priori knowledge, we consider our sparse Bayesian receiver where \(p_i\) and \(\rho_i^2\) are set to the values given in Table 10.1. These values were obtained by averaging over many realizations of the Saleh-Valenzuela channel. (Since no taps are exactly zero-valued in the Saleh-Valenzuela model, a tap was considered to be zero when its magnitude was less than 0.05 times the magnitude of the largest tap in the realization.) If a typical channel is available for measurement prior to the selection of the a priori parameters, this is the type of information that can possibly be obtained. To study the opposite case, i.e., when no a priori knowledge about the channel is available, we use our empirical Bayesian receiver. Note that this detector uses \(p_i = \hat{p}\) and \(\rho_i^2 = \hat{\rho}^2\) (i.e., \(p\) and \(\rho^2\) are the same for all taps). This results in a severely mismatched receiver (cf. Table 10.1 where the values of \(p_i, \rho_i^2\) vary rather heavily with \(i\)).

The receiver performances are shown in Figure 10.4. We also include the non-sparse Bayesian receiver where the channel has been truncated after \(n = 20\) taps in this figure: this is a relevant comparison since \(p_i \approx 1\) for \(i \geq 20\) (see Table 10.1). The Bayesian sparse receiver exploiting the full a priori knowledge from Table 10.1 is denoted “Bayesian sparse.” This receiver performs about 1 dB better than the non-sparse Bayesian receiver with \(n = 40\), and about
0.5 dB better than the non-sparse Bayesian receiver with \( n = 20 \). The empirical Bayesian receiver performs rather similarly to the sparse Bayesian receiver. One contributing reason for this is that the Bayesian receiver is very robust to mismatching \textit{a priori} knowledge (as indicated by Figures 10.1 and 10.2).

Table 10.1: The measured values of \( \{ \rho_i^2 \} \) and \( \{ p_i \} \) used for the “Bayesian sparse” curve in Figure 10.4

\[
\begin{align*}
\rho^2 &= [0.014124, 0.09741, 1.4791, 1.1895, 0.77271, 0.51005, \\
& \quad 0.33896, 0.23924, 0.17412, 0.12844, 0.10363, 0.086537, \\
& \quad 0.078362, 0.072019, 0.071404, 0.07419, 0.067682, \\
& \quad 0.066193, 0.061858, 0.056244, 0.046561, 0.044409, \\
& \quad 0.03711, 0.032066, 0.030457, 0.025896, 0.022456, \\
& \quad 0.021203, 0.020268, 0.017297, 0.014361, 0.01396, \\
& \quad 0.012081, 0.010978, 0.010211, 0.0093643, 0.0081263, \\
& \quad 0.007429, 0.0067383, 0.0066416] \\
\rho &= [0.835, 0.1767, 0.0044, 0.0119, 0.0222, 0.035, 0.0519, \\
& \quad 0.0698, 0.1036, 0.1399, 0.2016, 0.286, 0.3888, 0.5171, \\
& \quad 0.6101, 0.6956, 0.7405, 0.7623, 0.7787, 0.7858, 0.7946, \\
& \quad 0.7998, 0.8102, 0.8195, 0.8324, 0.8467, 0.8543, 0.8666, \\
& \quad 0.879, 0.8923, 0.8989, 0.9124, 0.9139, 0.9266, 0.9327, \\
& \quad 0.9372, 0.9488, 0.9563, 0.9642, 0.9682]
\end{align*}
\]
Figure 10.4: Saleh-Valenzuela channel model. The a priori knowledge from Table 10.1 was used by “Bayesian sparse”. A training sequence of length $n_t = 5$ was used. The closeup shows the region $\sigma^2 \in [-5, -7.5]$ dB.
10.6 Concluding Remarks

We have derived the optimal Bayesian RAKE receiver under the assumption of a sparse channel impulse response. We also derived an empirical Bayesian version of our receiver, which does not require any \textit{a priori} knowledge, nor the choice of any user parameters. The sparsity assumption was motivated both by physical considerations and by the fact that some existing channel models (e.g., for indoor radio) tend to generate sparse impulse responses. Via numerical simulations we demonstrated that our new receiver, and its empirical version, can significantly outperform the non-sparse Bayesian receiver when the channel is sparse. Our receiver has about the same complexity as MRC with channel estimation, which is the conventional solution.
10. RAKE Receiver for Channels with a Sparse Impulse Response
Chapter 11

Estimation of Semi-Sparse Radar Range Profiles

We study the use of semi-sparse models, i.e., models having a few coefficients that are significantly larger than the rest, for estimation of range profiles in radar and other active sensing applications. The estimation of such range profiles is equivalent to the estimation of a vector of regression coefficients in an underdetermined linear system. Each coefficient corresponds to a certain range bin in the illuminated area. If a range bin contains a target, the reflections from that bin will, in some applications, result in a value of the corresponding coefficient which is significantly larger than the value corresponding to a target-free range bin.

Under the assumption of a mixture of semi-sparse linear Gaussian models, we derive the MMSE (minimum mean square error) estimate of the range profile. We then find computationally efficient approximations of this MMSE estimate. We also derive a MAP (maximum a posteriori) target detector, that does not require the choice of any detection threshold. The performances of our methods are illustrated via numerical examples.

11.1 Introduction

An important problem in active sensing applications is to detect the locations of scatterers in a medium, such as free space, air, water, human body, etc. To achieve this goal, a directed beam is transmitted towards an area of interest where scatterers reflect it back to a receiver. The reflected signal contains information not only on the locations of the scatterers (associated with the delays of the reflected signals), but also on their respective “sizes”, or cross-
sections [139] (proportional to the strength of the reflected signals). The ranges of the scatterers and their respective strengths constitute a range profile. This range profile, in its discretized form, is a vector where each entry represents an area element. The magnitude of a vector element describes the reflection strength of the scatterers within the corresponding area. In this chapter, we are concerned with the problem of estimating range profiles as accurately as possible. We also obtain a target detection algorithm, which does not require the choice of any detection threshold.

In many applications, the medium is relatively homogeneous and contains only a few large objects. If this is the case, the resulting range profile will often be sparse or semi-sparse, i.e., most elements of the profile will be zero or nearly zero, while the remaining few elements will be significantly larger. In other words, the range profile will be such that the clutter level\(^1\) is low compared to the reflections from targets. Radar applications, for detecting aircrafts, marine vessels or objects in space [139], constitute the main motivation for our work. However, the methods we will present are also suitable for, e.g., ultrasound scans and non-destructive testing [104; 105].

More specifically, the methods we will present can be used when there are two distinct types of scatterers in the range profile: one target type (strong scatterers), and one clutter type (weak scatterers). The strictly sparse case, i.e., when an element is exactly zero with a non-zero probability, has been extensively studied; see, e.g., [22; 34; 45; 84] and the references therein. Strict sparsity is a special case of the more general semi-sparsity that we study in this chapter. In a semi-sparse model, an element is either close to zero or significantly larger. Semi-sparse models have been studied previously, e.g., in ultrasonics [104]. For simplicity, we allow for a slight abuse of terminology and sometimes use the term “sparse” for both the strictly sparse and the semi-sparse cases.

In this chapter we present an approximate MMSE estimator of the range profile under the assumption that it can be well modeled by a mixture of semi-sparse Gaussian linear models (similar to the Bernoulli-Gaussian model [79]). Our method is based on a method for MMSE estimation of strictly sparse models that was derived in [84] (see, Chapter 9). As will be shown, unlike some existing methods (see Sections 11.2.1 and 11.2.2), our estimator does not have a problem with sidelobes or smearing, and it does not rely on the assumption that the strong scatterers are present only in a specific interval. From our algorithm we also obtain the MAP (maximum a posteriori) target detector, assuming that the targets are independent, solitary scatterers.

The rest of the chapter is organized as follows: In Section 11.2 we describe the signal model together with some non-sparse range profile estimators. In Section 11.3 we present our semi-sparse MMSE estimator and MAP detector together with computationally efficient approximations. In Section 11.4 we

\(^1\)Clutter is a general term used for reflections which are not of interest, and the definition of what is clutter and what is not will vary from application to application. In the present chapter we are not concerned with the question as to which scatterers are “of interest” and which are not. We will simply use the term target for scatterers having large magnitudes, and clutter for scatterers having small magnitudes.
perform a numerical study and compare our estimators with some well-known methods. Finally, in Section 11.5, we summarize and discuss the results.

11.2 Signal Model and Previous Work

Consider a signal $s = [s(1) s(2) \cdots s(N)]^T \in \mathbb{C}^N$ transmitted towards a region of interest. The signal will be reflected against objects in the region, and the backscattered signal can be measured at the receiver.

Let the illuminated area be described by the unknown range profile $x = [x(1) x(2) \cdots x(n)]^T \in \mathbb{C}^n (n > N)$. Then the backscattered signal $y = [y(1) y(2) \cdots y(n - N + 1)]^T$ can be described by a convolution between the transmitted signal $s$ and the range profile [14]. In matrix form, this convolution becomes

$$y = \begin{bmatrix}
    s(N) & \cdots & s(1) & 0 & \cdots & \cdots & 0 \\
    0 & s(N) & \cdots & s(1) & \vdots & \ddots & \vdots \\
    \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
    \vdots \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
    0 & \cdots & \cdots & 0 & s(N) & \cdots & s(1)
\end{bmatrix} x + e, \quad (11.1)$$

where $e \in \mathbb{C}^{n-N+1}$ is a noise term. Note that (11.1) is an underdetermined linear regression equation; i.e., there are fewer equations $(n - N + 1)$ than there are unknowns $(n)$.

Also, it is important to note that the first and last $N - 1$ columns in $S$ do not contain the full transmitted signal $s$. For this reason, the corresponding first and last $N - 1$ elements in $x$ will be more difficult to estimate than the other elements, since they will be associated with lower transmitted energy in the measured backscattered signal $y$. The central elements, i.e., the elements in $\bar{x} = [x(N) x(N + 1) \cdots x(n - N + 1)]^T \in \mathbb{C}^{n-2(N-1)}$ are sometimes said to lie within the processing window, and the elements not in $\bar{x}$ are then of less interest[^2] [14]. Note that our MMSE estimator will be able to estimate the elements that are not in $\bar{x}$, although with less precision than that of the other elements. However, other methods may have more serious problems caused by the elements in $x$ that are not in $\bar{x}$; see below for details. For a fair comparison of the methods in the numerical simulations in Section 11.4, our focus will therefore be on the performance for estimation of $\bar{x}$.

11.2.1 Matched Filter Estimate

Perhaps the most commonly used method for range profile estimation is the so-called matched filter (MF) approach. The MF can be shown to maximize the

[^2]: Even so, it is important to include these elements in the model as they may contain large targets that can give a large contribution to $y$; see Sections 11.2.2 and 11.4.
signal-to-noise ratio (SNR) of a solitary point scatterer in additive white noise [139]. The MF estimate is obtained by correlating the received signal with the transmitted signal. For a solitary point scatterer in a noise-free environment, this operation gives the autocorrelation of the transmitted signal centered at the position of the scatterer and scaled by the corresponding reflection coefficient value. Ideally, the said autocorrelation sequence \( r(k) \) should resemble an impulse at lag \( k = 0 \). In this ideal case the MF output finds the point scatterer at the correct range, and there are no sidelobe effects caused by the nearby range elements. MF estimation of \( x \) is often called pulse compression [139] because of this goal of obtaining an impulse-like autocorrelation sequence. In this manner, the (approximate) impulse response of the range profile can be obtained even if a long signal is transmitted instead of just a pulse. This enables an increase of the transmitted energy, while maintaining a low peak-to-average power level, without unduly compromising the range resolution.

In practice, a pure impulse in the autocorrelation domain can never be realized. Consequently, a notable disadvantage of the MF estimate are the autocorrelation sidelobes. Large scatterers that leak through such sidelobes can mask smaller scatterers, thereby reducing their probability of detection [14]. Furthermore, in high resolution applications, a large single object in the area of interest may contribute to several adjacent elements in \( x \). The reflected signal can then be used to identify the object, since different objects will have different “signatures”. Such objects will, however, be subjected to smearing in the MF range profile estimate due to the non-zero sidelobes. This smearing can make the identification a difficult task. In Figure 11.1 we show, as an illustration, the autocorrelation sequence of the \( N = 30 \) Lewis-Kretschmer P3 code (11.19), see [89], that we will use in the numerical examples in Section 11.4.

The (normalized) MF estimate is given by the following expression:

\[
\hat{x}_{MF} = S^H y ÷ \text{diag}(S^H S)
\]  

(11.2)

where ÷ stands for element-wise division, and \( \text{diag}(S^H S) \) is the vector on the diagonal of the matrix \( S^H S \). Note that the elements on the edges of \( x \) are estimated using only a portion of the signal vector \( s \), which means that pulse compression may not work for them. This is so because the autocorrelation function of a portion of \( s \) does not necessarily resemble an impulse. Note also that, for the elements in \( \hat{x} \), the normalization factor in (11.2) becomes the more common \( s^H s \) [14].

11.2.2 Least Squares Estimate

Another common method for estimating \( x \) is the LS (least squares) method. It has been used in active sensing applications such as radar pulse compression, ultrasonic non-destructive testing, biomedical imaging, etc.; see [14] and the references therein. However, the ordinary LS estimate cannot be used for (11.1) directly. This is because there are fewer equations than unknowns in (11.1) which results in an infinite number of perfect LS fits. Therefore, it is often assumed that the first and last \( N - 1 \) elements of the range profile \( x \) are
Figure 11.1: Magnitude of the autocorrelation sequence $r(k)$ of the $N = 30$ Lewis-Kretschmer P3 code (11.19).
equal to zero. Under this assumption, the first and last \( N - 1 \) columns of \( S \) can be omitted from (11.1), and we obtain the following so-called LS model [14]:

\[
y = \begin{bmatrix} s(1) & 0 & \cdots & \cdots & 0 \\ \vdots & s(1) & 0 & \cdots & \vdots \\ s(N) & \vdots & \ddots & \ddots \\ 0 & s(N) & \ddots & \ddots & \vdots \\ \vdots & \ddots & s(1) & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & s(N) \end{bmatrix} \bar{x} + e
\]

(11.3)

where \( \bar{x} \) was defined earlier in this section. The LS estimator of \( \bar{x} \), which coincides with the ML (maximum likelihood) estimator if \( e \) is white, Gaussian and zero mean [75], is given by

\[
\hat{\bar{x}}_{LS} = \left( S_{LS}^H S_{LS} \right)^{-1} S_{LS}^H y.
\]

(11.4)

11.2.3 Non-Sparse MMSE Estimate

Instead of the truncation done to get the LS model (11.3) one can regularize the underdetermined system (11.1), e.g., by using a Gaussian prior for \( x \). If the prior \( x \sim \mathcal{CN}(0, \rho^2 I) \) is used, and the noise is \( e \sim \mathcal{CN}(0, \sigma^2 I) \), then the non-sparse MMSE (minimum mean square error) estimator of \( x \) is the following [75]:

\[
\hat{x}_{n-MMSE} = \left( S^H S + \frac{\sigma^2}{\rho^2} I \right)^{-1} S^H y.
\]

(11.5)

When all elements in the range profile have similar magnitudes, the non-sparse MMSE estimate is appropriate, provided that a good value for the SNR, \( \rho^2/\sigma^2 \), is available, and provided that the Gaussian assumptions on \( x \) and \( e \) are reasonable. However, for the semi-sparse problem we consider herein, we advocate the use of the method described in the next section instead of the non-sparse MMSE method.

11.2.4 Reiterative MMSE Approaches

In [14] and [81] two reiterative MMSE approaches for range profile estimation were introduced and discussed. The methods are reiterative in the sense that the unavailable a priori parameters required for the computation of the MMSE estimates of \( x \) (such as the variances of the elements in \( x \)) are first initialized and then iteratively updated using results from the previous iteration. In this manner approximate range profile MMSE estimates are obtained. The method

\( ^3 \)This should be seen as an approximation, as these elements will hardly be exactly zero-valued in practice.
we will present below treats the problem that the variances of the elements in \( x \) are unavailable in quite a different manner; namely by averaging over a set of models with different variance configurations in a multi-model fashion. The method in [14] is \textit{local} in the sense that each element in the range profile is treated separately, by replacing the matched filter by an adaptive filter which tries to “null” targets in other elements of the range profile. The method in [81] is \textit{global} in the sense that it estimates all range profile elements jointly, similarly to our approach below.

### 11.3 Semi-Sparse MMSE Estimate

In [84] an approximation of the MMSE estimator for a mixture of strictly sparse, linear and real-valued Gaussian models was derived (see, Chapter 9). Herein we extend that estimator in the following ways:

- We assume that the data is complex-valued; in [84] the data was assumed to be real-valued.
- We allow the variances of the elements in \( x \) to be different; in [84] the variances of all non-zero coefficients in \( x \) had the same value.
- We model the more general semi-sparse case, i.e., there are two candidate variances, that can take on any non-negative value, for each element in \( x \); in [84] the strictly sparse case was modeled, i.e., one of the two candidate variances was fixed to zero.
- We decrease the computational complexity by one or two orders of magnitude compared to [84].

We assume that the noise is white, circular complex Gaussian with zero mean and variance \( \sigma^2 \): \( e \sim \mathcal{CN}(0, \sigma^2 I) \). We also assume that the elements in \( x \) are independent and circular complex Gaussian zero mean random variables, \( x(j) \sim \mathcal{CN}(0, \gamma_j^2) \), \( j = 1, \ldots, n \), where \( \gamma_j^2 \) can take one of two possible values, with given \textit{a priori} probabilities:

\[
\gamma_j^2 = \begin{cases} 
\rho_j^2 & \text{target variance, a priori probability } P(\gamma_j^2 = \rho_j^2) = 1 - p_j \\
\nu_j^2 & \text{clutter variance, a priori probability } P(\gamma_j^2 = \nu_j^2) = p_j 
\end{cases}
\]

(11.6)

Without loss of generality we assume that \( \rho_j^2 > \nu_j^2 \). Since each of the \( n \) variances can take two different values (corresponding to target or clutter) we obtain \( 2^n \) possible models for \( x \). This quickly becomes an infeasible number of models to handle, so clearly we need some way of reducing this set. This aspect will be further discussed below.

The MMSE estimate of \( x \) equals the conditional mean:

\[
\hat{x}_{\text{MMSE}} = E[x|y] = \sum_{k=1}^{2^n} P(M_k|y)E[x|y, M_k]
\]

(11.7)
where \( M_k \) denotes a model\(^4\) for \( x \), associated with a specific (and unique; \( M_k \neq M_j \) if \( j \neq k \)) combination of target and clutter variances from (11.6), \( P(M_k | y) \) is the posterior probability of \( M_k \), and \( E[\cdot] \) denotes the expectation operator. Using Bayes’ rule, and considering only the models with indices in a given integer set \( \Omega \) (\( \Omega \) will be specified later and will consist of \( n + 1 \ll 2^n \) elements) we obtain the following approximation of the MMSE estimate [84]:

\[
\hat{x}_{\text{MMSE}} \approx \hat{x}_{\text{MMSE}} \triangleq \frac{\sum_{k \in \Omega} P(M_k)p(y|M_k)E[x|y,M_k]}{\sum_{l \in \Omega} P(M_l)p(y|M_l)}, \tag{11.8}
\]

where the omitted models are assumed to give a comparatively small contribution to \( \hat{x}_{\text{MMSE}} \). To compute \( \hat{x}_{\text{MMSE}} \) above we need to specify \( \Omega \) and the \textit{a priori} probabilities \( \{P(M_k)\}_{k \in \Omega} \). We also need expressions for \( p(y|M_k) \) and \( E[x|y,M_k] \). For this purpose we introduce the notation \( \Gamma_k = \text{diag}(\gamma_1^2, \ldots, \gamma_n^2) \), where the values of \( \{\gamma_j^2\}_{j=1}^n \) depend on the model \( M_k \).

From (11.6) we can compute the \textit{a priori} probabilities \( \{P(M_k)\}_{k \in \Omega} \) as

\[
P(M_k) = \left( \prod_{j \in \Psi_k} (1 - p_j) \right) \left( \prod_{j \in \{1, \ldots, n\} \backslash \Psi_k} p_j \right), \tag{11.9}
\]

where the set \( \Psi_k \) contains the indices of the elements for which \( \gamma_j^2 = \rho_j^2 \) (i.e., the target elements) under the model \( M_k \). Note that, since we consider only a subset \( \{M_k\}_{k \in \Omega} \) of all possible models, the corresponding \textit{a priori} probabilities (11.9) will not sum up to one. This is however of no serious concern to us; what is important is that the weighting terms in (11.8), i.e., \( P(M_k)p(y|M_k)/\sum_{l \in \Omega} P(M_l)p(y|M_l) \), sum up to one for \( k \in \Omega \), which they do.

Since both the noise \( e \) and the coefficient vector \( x \) are Gaussian we have that

\[
p(y|M_k) = \frac{1}{\pi^{n-N+1} |Q_k|} \exp \left( -y^H Q_k^{-1} y \right) \tag{11.10}
\]

where, if we let \( s_j \) denote the \( j \)-th column of \( S \), the data covariance matrix is

\[
Q_k = \sum_{j=1}^n \gamma_j^2 s_j s_j^H + \sigma^2 I = S \Gamma_k S^H + \sigma^2 I. \tag{11.11}
\]

Also, due to \( x \) and \( y \) being jointly Gaussian [84; 140]:

\[
E[x|y,M_k] = \Gamma_k S^H Q_k^{-1} y. \tag{11.12}
\]

To evaluate (11.8), what remains is to specify the \textit{a priori} parameters \( \{p_j, \rho_j^2, \nu_j^2\}_{j=1}^n \) and \( \sigma^2 \), and the set \( \Omega \). In Section 11.3.4 we discuss the choice of \textit{a priori} parameters. The set \( \Omega \) is chosen by successive model “reduction”, or backward elimination\(^5\) [165], as explained below.

\(^4\) We will not explicitly spell out the form of the models \( \{M_k\}_{k=1}^{2^n} \), as it is not necessary for our presentation.

\(^5\) Successive model reduction, or backward elimination, normally refers to reducing the complexity of the model, i.e., removing coefficients from consideration. Herein, we do not deal with explicit model reduction, unless \( \nu_j^2 = 0 \). Rather, the “reduction” refers to a possible target being removed from consideration, i.e., being assumed to be clutter.
11.3.1 Algorithm for Finding $\hat{x}_{\text{MMSE}}$

In this subsection we outline an algorithm for calculating $\hat{x}_{\text{MMSE}}$. We start by assuming that $\gamma_j^2 = \rho_j^2$ for $j = 1, \ldots, n$, i.e., that all elements in $x$ represent targets. Then, we successively replace target variances by clutter variances. Our starting model is referred to as the all-target model (the opposite one being the all-clutter model).

Since $s$ is known beforehand and $\sigma^2$, $\{\rho_j^2\}_{j=1}^n$ are a priori parameters we assume that $Q^{-1}$ and $|Q|$ for the all-target model are known (they can be pre-computed, and even if they are not initially available they can be computed in a fast manner; see [84]).

The algorithm for computing $\hat{x}_{\text{MMSE}}$ runs as follows:

1. Let $\mathcal{M}_c$ denote the current model which initially is set to the all-target model. Let $Q_c$ denote the corresponding data covariance matrix (11.11).
2. Compute $p(y|\mathcal{M}_c)$ and $E[x|y, \mathcal{M}_c]$ using (11.10) and (11.12). Store the results.
3. Perform one of the following two steps:
   
   (a) Let $\mathcal{M}_k$ be the model obtained by replacing one of the target variances in $\mathcal{M}_c$ by the corresponding clutter variance. Find the replacement that results in the largest $P(\mathcal{M}_k|y)$. It is sufficient to compute $p(y|\mathcal{M}_k)P(\mathcal{M}_k)$ for this task since $P(\mathcal{M}_k|y) = p(y|\mathcal{M}_k)P(\mathcal{M}_k)/p(y)$ where $p(y)$ does not depend on $\mathcal{M}_k$. Store the index of the corresponding variance in $j$.
   OR

   (b) Let $j$ be the index of the target variance in $\mathcal{M}_c$ for which the magnitude of the corresponding element in $E[x|y, \mathcal{M}_c]$ is minimum.$^6$
4. Update $\mathcal{M}_c$ by setting $\gamma_j^2 := \nu_j^2$. Compute new values for $Q_c^{-1}$ and $|Q_c|$.
5. If $\gamma_j^2 = \nu_j^2$ for $j = 1, \ldots, n$ (i.e., if we have arrived at the all-clutter model):

   Compute $p(y|\mathcal{M}_c)$ and $E[x|y, \mathcal{M}_c]$ using (11.10) and (11.12), and store the results. Using the results stored during the execution of the algorithm, compute $\hat{x}_{\text{MMSE}}$ from (11.8) and terminate.

   Otherwise, go to Step 2.

The proposed algorithm will select one model for each sparsity level, i.e., one model with $n$ targets, one model with one clutter element and $n-1$ targets, and so on down to the all-clutter model. This selection process results in $n+1$ models, and hence $\Omega$ in (11.8) consists of the indices of these $n+1$ models, which is much less than the $2^n$ models required for the exact version (11.7).

If the range profile is expected to be very sparse, we can consider several models for each of the lower sparsity levels. For example, when the current

---

$^6$This may not be appropriate when $\rho_j^2$ or $\nu_j^2$ vary a lot with $j$. 

---
model has become “sparse enough” one might consider selecting \( m > 1 \) target variances in Step 3 and continue the model reduction in separate “branches” (Step 4 is then performed for \( m \) different models), which will branch further at the next iteration, etc. Such an enhancement is straightforward to implement. However, for simplicity of exposition, we will not consider it herein.

Note, from (11.11), that replacing \( \rho_j^2 \) by \( \nu_j^2 \), which is done in Step 3a and Step 4, leads to the following update of \( Q_c \):

\[
Q_c := Q_c - (\rho_j^2 - \nu_j^2)s_js_j^H.
\]
(11.13)

Therefore, \( Q_c^{-1} \) and \( |Q_c| \) can be efficiently updated by using the matrix determinant lemma and the matrix inversion lemma [18; 147]:

\[
|Q_c| := (1 - (\rho_j^2 - \nu_j^2)s_j^H Q_c^{-1} s_j)|Q_c| \quad (11.14)
\]

\[
Q_c^{-1} := Q_c^{-1} - Q_c^{-1}s_j(Q_c^{-1}s_j)^H s_j^H Q_c^{-1} s_j - 1/(\rho_j^2 - \nu_j^2). \quad (11.15)
\]

Successively replacing clutter by targets starting from an all-clutter model, akin to the process of forward selection [165], is equally possible. In fact, by such an approach one can likely reduce the number of loops in the algorithm (i.e., reduce the number of elements in \( \Omega \)). This is so because, whenever the number of targets is small, the “target dense” models can be eliminated from \( \Omega \) without affecting \( \hat{x}_{\text{MMSE}} \) significantly. The reason why we preferred to present the backward elimination version of the algorithm is that it appears more safe to start with the least restrictive model in the set (i.e., the all-target model, which has the largest possible variances) as opposed to the most restrictive model (where all variances are set to the minimum possible values). Indeed, underestimation of a variance is often more harmful for inference than over-estimation of it [132]. By this reasoning, the forward selection version may have difficulties finding the appropriate clutter/target replacement. Even so, because of the attractive potential of complexity reduction, we will consider the forward selection approach in the numerical simulations in Section 11.4. Since the corresponding changes of the algorithm are straightforward, we omit spelling them out explicitly.

A free C++ implementation of our algorithms is available at the following web address:

http://www.it.uu.se/research/syscon/signalprocessing/SemiSparse.

This is also the implementation that we use for our numerical examples in Section 11.4.

### 11.3.2 Target Detection

The estimated range profile, \( \hat{x} \), is often used for detection of targets. Specifically, if we assume that the scatters are solitary and independent, elements in \( \hat{x} \) that have magnitudes greater than a certain threshold \( \tau \) indicate the existence of targets. The value of the threshold will affect the detection probability (the probability that a range bin containing a target is identified as such) and the probability of false alarm (the probability that a clutter element is falsely
detected as a target). A low threshold gives a high detection probability, but also a high probability of false alarm, and vice-versa [160]. In Appendix C we describe how to select the threshold, under some simplifying assumptions, such that the total probability of error is minimized.

An alternative approach for performing detection relies on selecting the MAP model:

\[
\mathcal{M}_{\text{MAP}} = \arg \max_{\mathcal{M}_k, k \in \{1, \ldots, 2^n\}} P(\mathcal{M}_k | y) = \arg \max_{\mathcal{M}_k, k \in \{1, \ldots, 2^n\}} p(y | \mathcal{M}_k) P(\mathcal{M}_k).
\]

![Equation 11.16](image)

The structure of the MAP model then tells us which range bins contain targets and which contain clutter: the elements in \(\hat{\mathcal{M}}_{\text{MAP}}\) for which \(\gamma_j^2 = \rho_j^2\) contain targets, and the elements for which \(\gamma_j^2 = \nu_j^2\) contain clutter. Similarly to the estimation task, considering all \(2^n\) models is infeasible, so instead we use the reduced set \(\Omega\):

\[
\hat{\mathcal{M}}_{\text{MAP}} = \arg \max_{\mathcal{M}_k, k \in \Omega} P(\mathcal{M}_k | y) = \arg \max_{\mathcal{M}_k, k \in \Omega} p(y | \mathcal{M}_k) P(\mathcal{M}_k).
\]

![Equation 11.17](image)

Because the algorithm in Section 11.3.1 stores the values of \(\{p(y | \mathcal{M}_k)\}_{k \in \Omega}\) and also computes \(P(\mathcal{M}_k)\), detection via (11.17) can be performed jointly with the estimation of \(x\) at very little extra cost.

### 11.3.3 Computational Complexity

We assume that the updates in (11.14) and (11.15) are used by the algorithm. Each of these updates requires \(O(M^2)\) operations, where \(M \triangleq n - N + 1\) denotes the number of rows in \(S\), as opposed to \(O(M^3)\) if \(|Q|\) and \(Q^{-1}\) would be computed directly.

Step 3a requires \(n - l\) computations of \(Q^{-1}\) and \(|Q|\), where \(l\) denotes the current loop number in the execution of the algorithm (starting with \(l = 0\)). These computations will dominate the computational load of the algorithm, and since Step 3a needs to be performed \(n - 1\) times in total, the algorithm employing Step 3a will have a computational load of \(O(n^2 M^2)\). This should be compared to the implementation in [84], which has a computational load of \(O(n^3 M^2)\). Note, though, that the computations of \(Q^{-1}\) and \(|Q|\) in Step 3a can be implemented in parallel, thereby reducing the load of this version of the algorithm to \(O(n M^2)\), provided that \(n\) processors are available.

The version of the algorithm that employs Step 3b need only perform one computation of \(Q^{-1}\) and \(|Q|\) at each loop of the algorithm. Therefore, the complexity of this version is \(O(n M^2)\).

Both versions of the algorithm can be speeded up by truncating the set \(\Omega\) even further (similarly to what can be done for the forward selection version of the algorithm; see above). For instance, the starting model can use target variances only for the \(K < n\) largest values of the \(\hat{x}_{\text{MAP}}\) (letting the rest be clutter). This will result in \(K < n\) model reduction steps, i.e., \(K\) loops of the algorithm. In this manner, the computational complexities are reduced to \(O(K n M^2)\) for the version of the algorithm employing Step 3a, and \(O(K M^2)\) for the parallel Step 3a-version and the Step 3b-version.
11.3.4 The Model Versus Reality Issue

An important question to consider is whether the above signal model is a good description of reality in the application of interest. In the Introduction we have already discussed the issue of sparsity and noted that there are important applications for which the range profile can be expected to be sparse (although this does not hold for all active sensing applications). To motivate the assumption that the scatterers are Gaussian we note that, in accordance with the maximum entropy principle [69], the Gaussian assumption is the least restrictive one, and thus the safest assumption to make if no other information is available. If, in a specific application, another distribution is known to describe the clutter or the targets better, one should of course try to exploit that information. Finally, we note that the Gaussian assumption on the clutter holds in many applications. For instance, sea clutter can be well modeled by means of the Gaussian distribution [139].

Regarding the role of the a priori parameters, \( \{p_j, \sigma_j^2, \nu_j^2\}_{j=1}^n \) and \( \sigma^2 \), in the estimator, there are two important issues to consider. The first is the issue of robustness: We would like the approximate MMSE/MAP estimates (11.8), (11.17) to be robust against small variations in the parameters. Second, we need some knowledge regarding appropriate values of these parameters. In most applications we would probably set \( p_j = p, \sigma_j^2 = \sigma^2, \nu_j^2 = \nu^2 \) for \( j = 1, \ldots, n \), unless very specific a priori information were available. This reduced set of a priori parameters is what we will use in the numerical simulations in Section 11.4.

Information about \( p, \sigma^2, \nu^2 \) and \( \sigma^2 \) can possibly be obtained in an empirical Bayesian framework (i.e., the parameters can be estimated from the data) similar to what was done in [84]. One could, e.g., try to estimate the parameters based on the MF estimate (11.2). We choose, however, not to pursue such an approach here, for reasons of conciseness. Instead, we will assume that “reasonable values” for \( p, \sigma^2, \nu^2 \) and \( \sigma^2 \) are available and study the estimators’ robustness to variations in the parameter values.

11.4 Numerical Examples

In this section we present a number of numerical examples to illustrate the performance of various estimators. The comparison is partly based on the empirical MSEs (mean square errors) of the parameter estimates obtained by Monte Carlo simulation:

\[
\text{empirical MSE}(\hat{x}) = \frac{1}{L\bar{n}} \sum_{j=1}^{L} (\hat{x}_j - \bar{x}_j)^H(\hat{x}_j - \bar{x}_j),
\]

where \( \bar{x}_j \) is a sub-vector of \( x_j \), obtained in the \( j \)th Monte Carlo run, consisting of \( \bar{n} \) elements (we will study the target estimation performance and the clutter estimation performance separately), \( \hat{x}_j \) is the estimate of \( \bar{x}_j \), and \( L = 1000 \).

\(^7\)This is possible to do in the framework described here, since the algorithm is general in the sense that only the expressions related to the selected distributions need to be changed.
is the total number of Monte Carlo runs. For illustration purposes we also directly study (in Example 1, below) a single range profile realization and the estimates of it. We also consider a detection example.

In all examples presented here, the clutter elements are independently drawn from a circular Gaussian zero mean distribution with variance $\nu^2$. The targets elements, however, are sometimes set to deterministic values. This results in a mismatch between the data generating process and the assumptions used by the MMSE estimator. For the transmitted sequence $s$ we use the length $N = 30$ polyphase modulated Lewis-Kretschmer P3 code [89] which, after baseband conversion, is defined as [14]

$$s(k) = \exp \left( \frac{i\pi(k - 1)^2}{N} \right), \quad k = 1, \ldots, N. \quad (11.19)$$

The autocorrelation sequence of $s$ can be seen in Figure 11.1. The illuminated area, i.e., our range profile $\mathbf{x}$, is of length $n = 158$. This means that the part of the range profile that lies within the processing window, i.e., $\tilde{\mathbf{x}}$, is of length $\tilde{n} = 158 - 2(N - 1) = 100$.

We evaluate the following methods: (a) The MF estimator (11.2); (b) the LS estimator (11.4); (c) the non-sparse MMSE estimator (11.5), denoted as n-MMSE; (d) the sparse approximate MMSE estimator employing Step 3a in the algorithm described in Section 11.3.1, denoted as MMSE 3a; (e) the sparse approximate MMSE estimator employing Step 3b in the algorithm described in Section 11.3.1, denoted as MMSE 3b; (f) the sparse forward selection approximate MMSE estimator described at the end of Section 11.3.1, employing the forward selection step corresponding to Step 3b, and aborting after having replaced $K = 10$ clutter elements by target elements, denoted as MMSE fsel $K = 10$. For the detection example (see Example 1) we use the approximate MAP estimates, described in Section 11.3.2, that correspond to the MMSE estimates (d), (e) and (f) above (i.e., (11.17) with the same $\Omega$ as in (d), (e) and (f) above). We denote these estimates as MAP 3a, MAP 3b and MAP fsel $K = 10$, respectively.

### 11.4.1 Example 1: Estimation and Detection with Closely Spaced Targets

Here we study the estimation performance when the range profile contains two closely spaced targets near the center of the profile. We also add a target just outside the processing window. The target elements are set to $x(28) = 1 + 1i$, $x(65) = 1$ and $x(69) = 0.2$. The remaining elements in $\mathbf{x}$ are clutter with variance $\nu^2 = -30$ dB. The noise level is set to $\sigma^2 = -30$ dB. The approximate MMSE estimators (and MAP estimators; see below) are supplied with the true values of $\sigma^2$ and $\nu^2$, and also with $p = 1 - 3/n$ and $\rho^2 = 0$ dB.

In Figure 11.2 we show an example realization of the magnitude of the range profile $\mathbf{x}$; the so-called ground truth. We also show the magnitudes of the MF estimate and the MMSE 3a estimate. As we see, the sidelobes of the larger scatterer overlap with the smaller scatterer in the MF estimate. Therefore, the MF estimate fails to resolve the peaks. The MMSE 3a estimate does not have
any problems to resolve the peaks; also, it is able to follow the range profile accurately, at least within the processing window.

The LS estimate, the non-sparse MMSE estimate, MMSE 3b and MMSE fsel are omitted from the Figure 11.2, for reasons of readability. Both the LS estimate and the non-sparse MMSE estimate are able to resolve the targets. However, the large peak just outside the processing window causes “ghost peaks” to appear in the LS estimate, i.e., broad peaks at locations where there are no targets (remember that the LS estimate relies on the assumption that the range profile is zero-valued outside the processing window). This phenomenon has been observed before [14]. The non-sparse MMSE estimate follows the ground truth more closely than the MF estimate does, but not as well as the MMSE 3a estimate. The MMSE 3b and the MMSE fsel estimates coincide with the MMSE 3a estimate in this case.

We also study the detection capabilities of the methods, i.e., how often they manage to correctly identify the target and clutter elements within the processing window. Again, we base our results on 1000 Monte Carlo runs. For this detection study we require a threshold $\tau$ for MF, LS and $n$-MMSE: a range element with index $j$ is said to contain a target when $|\hat{x}(j)| > \tau$, and to
11.4. Numerical Examples

contain clutter otherwise. We use the threshold value which gives the minimum probability of making an error when identifying an element as target or clutter. To compute this threshold (under some simplifying assumptions) we use the approach described in Appendix C, with the same a priori parameter values, \( p = 1 - 3/n \), \( \rho^2 = 0 \) dB and \( \nu^2 = -30 \) dB, as for the semi-sparse MMSE/MAP estimators. The threshold then becomes \( \tau \approx 0.1042 \). The MAP approach described in Section 11.3.2 does not depend on the choice of a threshold. For this approach, the model structure itself, \( \hat{\mathcal{M}}_{\text{MAP}} \), contains the information on which elements are targets (the elements for which \( \gamma_j^2 = \rho_j^2 \)) and which are clutter (the elements for which \( \gamma_j^2 = \nu_j^2 \)).

In Table 11.1 we show the probabilities of correct detection, i.e., the percentages with which the methods correctly identify the target elements in \( \hat{x} \) (we assume that we are interested in only the elements within the processing window). We also show, in Table 11.2, the false alarm rates, i.e., the percentages with which the methods falsely detect targets where there is clutter in \( \hat{x} \). In the tables, MAP 3a, MAP 3b and MAP fsel correspond to different approximations (different ways of selecting \( \Omega \)) of the MAP estimate (see Section 11.3.2).

We note that all methods have very high probability of correct detection when the noise level is low, with MAP 3b slightly below the rest. For very high noise levels, the threshold based methods MF, LS and n-MMSE have higher probabilities of correct detection than the MAP based methods. However, they also have much higher false alarm rates. The reason for this behavior is that MF, LS and n-MMSE tend to view most coefficients as targets when \( \sigma^2 \) is large, since the strong noise component then “leaks” through the estimator and causes the estimates to exceed the threshold with high probability. The MAP methods behave in a more intuitive way, with probabilities of correct detection that consistently decrease with increasing noise levels. They also maintain very low false alarm rates for all noise levels.

The detection performance can also be analyzed by considering the probability of correct identification of the entire processing window. By “correct identification”, we mean that all clutter elements and all target elements within the processing window have been correctly identified. This measure corresponds to “1 − the frame error rate” for communication systems. In Table 11.3 we show the percentages of correct identification for the methods under study.

We note that MF and LS completely fail to identify the range profiles correctly. The reason for this is the high false alarm rates (see Table 11.2) for these methods. The n-MMSE method performs better for low noise levels, since its false alarm rate then is low. The MAP methods show the overall best performance.

We also note that both MF and LS are expected to have problems in the present example: MF has problems to resolve closely spaced targets, and LS has problems when there are targets just outside the processing window. This difficult example was designed to illustrate that our MAP-based methods do not suffer from the above problems.
### Table 11.1: Probabilities of correct detection for Example 1.

<table>
<thead>
<tr>
<th>noise level $\sigma^2$ [dB]</th>
<th>MF</th>
<th>LS</th>
<th>n-MMSE</th>
<th>MAP 3a</th>
<th>MAP 3b</th>
<th>MAP fsel $K = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-60$</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>99.65</td>
<td>100.00</td>
</tr>
<tr>
<td>$-50$</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>98.60</td>
<td>100.00</td>
</tr>
<tr>
<td>$-40$</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>98.15</td>
<td>100.00</td>
</tr>
<tr>
<td>$-30$</td>
<td>98.35</td>
<td>98.85</td>
<td>99.25</td>
<td>95.10</td>
<td>57.95</td>
<td>75.15</td>
</tr>
<tr>
<td>$-20$</td>
<td>91.20</td>
<td>95.85</td>
<td>94.05</td>
<td>50.35</td>
<td>50.15</td>
<td>50.55</td>
</tr>
<tr>
<td>$-10$</td>
<td>98.65</td>
<td>98.75</td>
<td>97.25</td>
<td>6.20</td>
<td>6.00</td>
<td>6.20</td>
</tr>
</tbody>
</table>

### Table 11.2: False alarm rates for Example 1.

<table>
<thead>
<tr>
<th>noise level $\sigma^2$ [dB]</th>
<th>MF</th>
<th>LS</th>
<th>n-MMSE</th>
<th>MAP 3a</th>
<th>MAP 3b</th>
<th>MAP fsel $K = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-60$</td>
<td>7.34</td>
<td>12.74</td>
<td>0.04</td>
<td>0.01</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>$-50$</td>
<td>7.44</td>
<td>12.94</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$-40$</td>
<td>7.48</td>
<td>12.97</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$-30$</td>
<td>7.54</td>
<td>12.85</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$-20$</td>
<td>8.12</td>
<td>13.73</td>
<td>0.12</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$-10$</td>
<td>18.94</td>
<td>27.14</td>
<td>11.38</td>
<td>0.01</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>$0$</td>
<td>74.43</td>
<td>78.96</td>
<td>73.77</td>
<td>0.04</td>
<td>0.01</td>
<td>0.04</td>
</tr>
<tr>
<td>$10$</td>
<td>96.78</td>
<td>97.47</td>
<td>94.23</td>
<td>0.08</td>
<td>0.07</td>
<td>0.08</td>
</tr>
</tbody>
</table>

### Table 11.3: Probabilities of correct identification of the entire processing window (100% = “frame error rate”) for Example 1.

<table>
<thead>
<tr>
<th>noise level $\sigma^2$ [dB]</th>
<th>MF</th>
<th>LS</th>
<th>n-MMSE</th>
<th>MAP 3a</th>
<th>MAP 3b</th>
<th>MAP fsel $K = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-60$</td>
<td>0.0</td>
<td>0.0</td>
<td>96.2</td>
<td>99.4</td>
<td>98.9</td>
<td>99.8</td>
</tr>
<tr>
<td>$-50$</td>
<td>0.0</td>
<td>0.0</td>
<td>98.3</td>
<td>99.8</td>
<td>97.4</td>
<td>99.8</td>
</tr>
<tr>
<td>$-40$</td>
<td>0.0</td>
<td>0.0</td>
<td>98.2</td>
<td>99.7</td>
<td>97.1</td>
<td>99.7</td>
</tr>
<tr>
<td>$-30$</td>
<td>0.0</td>
<td>0.0</td>
<td>97.2</td>
<td>99.7</td>
<td>96.4</td>
<td>99.7</td>
</tr>
<tr>
<td>$-20$</td>
<td>0.0</td>
<td>0.0</td>
<td>88.4</td>
<td>99.7</td>
<td>82.9</td>
<td>99.7</td>
</tr>
<tr>
<td>$-10$</td>
<td>0.0</td>
<td>0.0</td>
<td>80.1</td>
<td>50.1</td>
<td>15.9</td>
<td>50.2</td>
</tr>
<tr>
<td>$0$</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>$10$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
11.4.2 Example 2: Estimation with Target Signature

In this example we model a high resolution application with a multi-element target near the center of the range profile: \( x(65), x(66), x(67), x(68), x(69) \) = \([1 + i, 0.5 - 0.5i, 0.2 + 0.9i, -0.6 + 1.2i, 0.6 - 0.4i]\). We also add a solitary scatterer outside the processing window: \( x(7) = 0.6 - 0.4i \). The remaining elements are clutter, with variance \( \nu^2 = -20 \text{ dB} \). The noise level \( \sigma^2 \) is varied from \(-60 \text{ dB}\) to \(0 \text{ dB}\). We supply the values of \( \nu^2 \) and \( \sigma^2 \), as well as the values \( \rho^2 = 0 \text{ dB} \) and \( p = 1 - 6/n \) to the estimators.

In Figure 11.3 we plot the empirical MSEs for the target elements within the processing window, i.e., we use \( \tilde{x} = [x(65) \cdots x(69)]^T \) in (11.18). In Figure 11.4 we plot the empirical MSEs for the clutter elements within the processing window\(^8\).

As can be seen, the MF performs poorly compared to the other methods. The reason is likely the sidelobes of the autocorrelation function of \( s \). LS performs surprisingly well, considering the fact that it ignores the target outside the processing interval. The non-sparse MMSE performs worse than LS in Figure 11.3. This is probably because of the mismatch between the data generating process and the assumptions used in n-MMSE. All the sparse MMSE estimates perform very well, and almost identically to one another. Note, however, that the MSEs of all estimates level out for low noise variances. This is due to a non-ideal pulse compression sequence, for MF, and due to mismatches between the data generating process and the assumptions, for the other methods.

11.4.3 Example 3: Estimation With a Priori Parameter Mismatch

In this example we study the behavior of the estimates when there are mismatches in the a priori parameters. We vary the assumed values for these parameters in a one-by-one fashion. The true range profiles have a solitary Gaussian distributed point scatterer: \( x(65) \sim \mathcal{CN}(0, \rho^2) \). The true values of the a priori parameters are set to \( \sigma^2 = -30 \text{ dB} \), \( \nu^2 = -20 \text{ dB} \) and \( \rho^2 = 0 \text{ dB} \). Note that we have one target element (i.e., a deterministic number of targets), so \( p \) is not a parameter used in the data generation process. However, one might expect that the best choice of \( p \) would be \( p = 1 - 1/n \approx 0.9937 \). We supply these values of the a priori parameters to the estimators in the examples below, except for the parameter value that is varied in each specific example.

In Figure 11.5 we show the MSEs of the estimated values of the target element (\( \tilde{x} = x(65) \) in (11.18)) when we vary the value of \( p \) supplied to the semi-sparse MMSE estimators. We observe from the figure, that the MMSE estimators appear to be very robust to the choice of \( p \) (the variations in the MMSE curves appear to be more due to different random realizations of \( x \) and \( e \) than due to the choice of \( p \)). The corresponding figure for the clutter elements is very similar (the MSEs increase with about 2 dB for all methods) and it is therefore omitted.

---

\(^8\)Even though the clutter elements themselves may not be of interest, their accurate estimation is needed for the detection of targets.
Figure 11.3: Example 2: Empirical MSEs for the target elements of $\hat{x}$ within the processing window.
Figure 11.4: Example 2: Empirical MSEs for the clutter elements of $\hat{x}$ within the processing window.
Figure 11.5: Example 3: Empirical MSEs for the target element of $\hat{x}$ when there is one target element (corresponding to $p = 1 - 1/158 \approx 0.9937$) and the assumed value of $p$ for the sparse MMSE estimators is varied. The inlay plot is a zoom.
Figure 11.6: Example 3: Empirical MSEs for the target element of $\hat{x}$ when $\rho^2 = 0$ dB and the assumed value of $\rho^2$ for the MMSE estimators is varied.

Figure 11.6 shows the MSEs for the target element when we vary the value of $\rho^2$ that we supply to the MMSE estimators. We note from the plot that the MMSE estimators continue to show good performance even for deviations of $\rho^2$ up to at least 15 dB from the true value of this parameter. Also note, from (11.5), (11.11) and (11.12), that the parameter $\rho^2$ affects $\hat{x}_{\text{MMSE}}$ and $\hat{x}_{\text{n-MMSE}}$ similarly. Therefore, one can realize from (11.5) that the choice of $\rho^2$ is not critical for $\hat{x}_{\text{MMSE}}$ and $\hat{x}_{\text{n-MMSE}}$ as long as $S^H S$ dominates over $\frac{\nu^2}{\rho^2} I$ (as in the present example). The MSEs for the clutter elements behave very similarly to the MSEs in Figure 11.6 (the MSEs increase with about 2 dB for all methods), so we omit the corresponding figure.

In Figure 11.7 we show the MSEs for the target element when we vary the value of $\nu^2$ that we supply to the semi-sparse MMSE estimators. As can be seen, even for relatively large variations, the sparse MMSE estimators outperform the other methods. The exception is for a choice of $\nu^2$ that is more than 10 dB below the true value: the MSEs of MMSE 3a and MMSE 3b then become larger than the MSEs of LS and n-MMSE. The MSEs for the clutter elements behave very similarly to the MSEs in Figure 11.7 (the MSEs increase with about 2 dB for all methods), so we omit the corresponding figure.

In Figure 11.8 we study the MSEs for the target element when we vary the value of $\sigma^2$ that we supply to the MMSE estimators. As can be seen
Figure 11.7: Example 3: Empirical MSEs for the target element of $\hat{x}$ when $\nu^2 = -20$ dB and the assumed value of $\nu^2$ for the sparse MMSE estimators is varied.
11.4. Numerical Examples

11.4.1 Example 3: Empirical MSEs

![Figure 11.8: Example 3: Empirical MSEs for the target element of $\hat{x}$ when $\sigma^2 = -30$ dB and the assumed value of $\sigma^2$ for the MMSE estimators is varied.](image)

From the figure, there is very little variation in the performance of the MMSE estimators as $\sigma^2$ varies, as long as $\sigma^2$ is not chosen much too large. Also here, the corresponding figure of the MSEs for the clutter elements is very similar to Figure 11.8, and it is therefore omitted.

11.4.4 Example 4: Average Computation Time

Finally, we study the average computation times for the estimators when the length of the range profile is varied. We use the same signal $s$ as before, of length $N = 30$. The length of the range profile $n$ is varied such that the processing window length, $\bar{n} = n - 2(N-1)$, varies between 10 and 100. In Figure 11.9 we show the average computation times for the different methods implemented in C++ when running on a dedicated 2.4 GHz AMD Opteron 250 processor (without any parallelization). The computation time is averaged over 1000 runs for each data point in the figure. Note that we pre-compute some expressions in the estimators: the expressions depending only on $s$ and on the a priori parameters can be computed prior to receiving $y$. This means that MF, LS and n-MMSE essentially boil down to a matrix-vector multiplication (everything except $y$ can be pre-computed), and that $Q^{-1}$ and $|Q|$ are available for the starting model of the semi-sparse MMSE methods.
As can be seen from the figure, MF, LS and n-MMSE are much faster than the other methods: in this example the semi-sparse estimators require at least 100 times more running time. The reason why LS is slightly faster than MF and n-MMSE is because of the truncation which means that only $\bar{x}$ is estimated; see Section 11.2.2. MF and n-MMSE, on the other hand, estimate the entire $x$ in our implementation. If we only estimate $\bar{x}$, MF and n-MMSE require computation times similar to that of LS.

If a low computation time is essential, then MF, LS or n-MMSE are likely the best choice. However, if the increase in computation time can be afforded, then the semi-sparse MMSE methods may be more appropriate, since they have much better detection and estimation performances than MF, LS and n-MMSE have. Note that, as discussed in Section 11.3.1, it is possible to derive faster versions of MMSE 3a and MMSE 3b which should require computation times similar to those of MMSE fsel (specifically, MMSE 3a can be implemented using parallel processors; see Section 11.3.3). The price payed for this decreased computation time is likely a slight decrease in detection and estimation performance.
11.5 Concluding Remarks

We have presented several versions of an approximate algorithm for MMSE estimation and MAP detection of a semi-sparse range profile. The semi-sparse MMSE/MAP estimators are able to outperform well-known estimators such as the MF, LS and the non-sparse MMSE. Specifically, our estimators do not have any problems with closely spaced targets, or targets outside the processing interval. The semi-sparse estimators were derived in a Bayesian framework, and they depend on some user parameters that describe the \textit{a priori} knowledge of the range profile under study. We have shown via numerical simulations that the semi-sparse MMSE estimators are robust to the choice of the user parameters, in the sense that the performance of the estimators is not seriously affected when these parameters depart from their true values used to generate the data. Furthermore, the different versions of the semi-sparse estimators used herein perform very similarly, which may be a reason to prefer the faster versions of the algorithm, viz. MMSE/MAP 3b or MMSE/MAP fsel.
11. Estimation of Semi-Sparse Radar Range Profiles
Part III
Appendices
Cyclic Minimizers, Majorization Techniques, and the Expectation-Maximization Algorithm

Appendix A

A.1 Introduction

Many parameter estimation problems in signal processing and elsewhere can be reduced to the task of minimizing a function of the unknown parameters. More often than not, this task is rather difficult owing to the existence of possibly many local minima and the sharpness of the global minimum. In this appendix we will review three approaches that can be used to minimize functions of the type encountered in parameter estimation problems. The first two approaches, viz. the cyclic minimization and the majorization technique, are quite general, whereas the third one, viz. the expectation-maximization (EM) algorithm, is tied to the use of the maximum likelihood method for parameter estimation.

The main goal of this appendix is to provide a quick introduction of the aforementioned approaches. To achieve this goal, the presentation is as brief, general and simple as possible throughout the appendix. Also, the discussion is focused on the “essentials” of each approach, and similarly the references are limited to a few general publications.

We will denote the function to be minimized by $f(\theta)$, where $\theta$ is a vector. Sometimes we will write this function as $f(u, v)$ where $[u^T, v^T]^T = \theta$. The algorithms for minimizing $f(\theta)$ discussed in this appendix are iterative. We let $\theta^i$ denote the value taken by $\theta$ at the $i$th iteration (and similarly for $u$ and $v$). The common feature of the algorithms included in this appendix is that they
all monotonically decrease the function at each iteration:

\[ f(\theta^{i+1}) \leq f(\theta^i) \quad \text{for } i = 0, 1, 2, \ldots \]  

(A.1)

Hereafter, \( \theta^0 \) denotes the initial value (or estimate) of \( \theta \) used by the minimization algorithm in question. Clearly (A.1) is an appealing property which in effect is the main reason for the interest in the algorithms discussed here. However, we should note that usually (A.1) can only guarantee the convergence to a local minimum of \( f(\theta) \). The goodness of the initial estimate \( \theta^0 \) will often determine whether the algorithm will converge to the global minimum. In fact, for some of the algorithms discussed in what follows, not even the convergence to a local minimum is guaranteed. For example, the EM algorithm can converge to saddle points (see, e.g., [95]). Such a behavior is rare in some applications, but in others it may be rather common [13].

A.2 Cyclic Minimizer

To describe the main idea of this type of algorithm in its simplest form, let us partition \( \theta \) into two subvectors:

\[ \theta = \begin{bmatrix} u \\ v \end{bmatrix}. \]  

(A.2)

Then the generic iteration of a cyclic algorithm for minimizing \( f(u, v) \) will have the following form:

\[
\begin{align*}
v^0 &= \text{given} \\
\text{For } i = 1, 2, \ldots \text{ compute:} \\
u^i &= \arg\min_u f(u, v^{i-1}) \\
v^i &= \arg\min_v f(u^i, v). 
\end{align*}
\]  

(A.3)

An obvious modification of (A.3) allows us to start with \( u^0 \), if so desired. Note that (A.3) alternates (or cycles) between the minimization of \( f(u, v) \) with respect to \( u \) for given \( v \) and the minimization of \( f(u, v) \) with respect to \( v \) for given \( u \), and hence the name of “cyclic” given to this type of algorithm. It is readily verified that the cyclic minimizer (A.3) possesses the property (A.1):

\[
f(u^i, v^i) \leq f(u^i, v^{i-1}) \leq f(u^{i-1}, v^{i-1})
\]

where the first inequality follows from the definition of \( v^i \) and the second from the definition of \( u^i \).

The partitioning of \( \theta \) into subvectors is usually done in such a way that the minimization operations in (A.3) (or, at least one of them) are “easy” (in any case, easier than the minimization of \( f \) jointly with respect to \( u \) and \( v \)). Quite often, to achieve this desideratum we need to partition \( \theta \) in more than two subvectors. The extension of (A.3) to such a case is straightforward and will not be discussed here. However, there is one point about this extension that we would like to make briefly: whenever \( \theta \) is partitioned in three or
more subvectors we can choose the way in which the various minimization subproblems are iterated. For instance, if \( \theta = [u^T, v^T, w^T]^T \) then we may iterate the minimization steps with respect to \( u \) and with respect to \( v \) a number of times (with \( w \) being fixed), before re-determining \( w \), and so forth.

### A.3 Majorization Technique

The main idea of this type of iterative technique for minimizing a given function \( f(\theta) \) is quite simple (see, e.g., [62] and the references therein). Assume that at the \( i \)th iteration we can find a function \( g_i(\theta) \) (the subindex \( i \) indicates the dependence of this function on \( \theta^i \)) which possesses the following properties:

\[
\begin{align*}
g_i(\theta^i) &= f(\theta^i) \\
g_i(\theta) &\geq f(\theta)
\end{align*}
\]

and

the minimization of \( g_i(\theta) \) with respect to \( \theta \) is “easy” (or, in any case, easier than the minimization of \( f(\theta) \)).

Owing to (A.5), \( g_i(\theta) \) is called a majorizing function for \( f(\theta) \) at the \( i \)th iteration. In the majorization technique, the parameter vector at iteration \((i + 1)\) is obtained from the minimization of \( g_i(\theta) \):

\[
\theta^{i+1} = \arg\min_{\theta} g_i(\theta).
\]

The key property (A.1) is satisfied for (A.7), since

\[
f(\theta^i) = g_i(\theta^i) \geq g_i(\theta^{i+1}) \geq f(\theta^{i+1}).
\]

The first inequality in (A.8) follows from the definition of \( \theta^{i+1} \) in (A.7), and the second inequality from (A.5). Note that any parameter vector \( \theta^{i+1} \) which gives a smaller value of \( g_i(\theta) \) than \( g_i(\theta^i) \) will satisfy (A.8). Consequently, whenever the minimum point of \( g_i(\theta) \) (see (A.7)) cannot be derived in closed-form we can think of determining \( \theta^{i+1} \), for example, by performing a few iterations with a gradient-based algorithm initialized at \( \theta^i \) and using a line search (to guarantee that \( g_i(\theta^{i+1}) \leq g_i(\theta^i) \)). We should note that a similar observation could be made on the cyclic minimizer in (A.3) when the minimization of either \( f(u, v^{i-1}) \) or \( f(u^i, v) \) cannot be done in closed-form. Note, however, that the modification of either (A.7) or (A.3) in this way, with the goal of simplifying the computational effort of each iteration, may slow down the convergence speed of the algorithm by increasing the number of iterations needed to achieve convergence.

We remark on the fact that the cyclic minimizer requires the user to choose a partitioning of \( \theta \) that makes the minimization in, e.g., (A.3) “easy”, whereas the majorization technique requires a function \( g_i(\theta) \) that is not only “easy” to
minimize but also possesses the essential property (A.5). Fortunately for the majorization approach, finding such functions \( g_i(\theta) \) is not as hard as it might seem. Indeed, in the next section we will review a method for constructing a function \( g_i(\theta) \) possessing the desired properties (A.4) and (A.5) for a general class of functions \( f(\theta) \) that are commonly used in parameter estimation exercises.

## A.4 EM Algorithm

The maximum likelihood (ML) is probably the most widely studied method of parameter estimation. In what follows we assume that this is the method used for parameter estimation, and hence that the function we want to minimize with respect to \( \theta \) is the negative log-likelihood:

\[
f(\theta) = -\ln p(y|\theta)
\]

(A.9)

where \( p(y|\theta) \) is the probability density function of the data vector \( y \) conditioned on the parameter vector \( \theta \). Our main goal in this section is to show how to construct a majorizing function for the estimation criterion in (A.9) and how the use of the corresponding majorization technique leads to the expectation-maximization (EM) algorithm introduced in [28] (see also [95] and [99] for more recent and detailed accounts on the EM algorithm; other references with a signal processing-related flavor include [24; 37; 38]).

A notation that will be frequently used in the following concerns the expectation with respect to the distribution of a certain random vector, let us say \( z \), which we will denote by \( E_z\{\cdot\} \). When the distribution concerned is conditional on another random vector, let us say \( y \), we will use the notation \( E_{z|y}\{\cdot\} \). If we also want to stress the dependence of the distribution (with respect to which the expectation is taken) on a certain parameter vector \( \theta \), then we write \( E_{z|y,\theta}\{\cdot\} \).

The main result which we will use in the following is **Jensen’s inequality.** It asserts that for any concave function \( h(x) \), where \( x \) is a random vector, the following inequality holds:

\[
E\{h(x)\} \leq h(E\{x\}) .
\]

(A.10)

The proof of (A.10) is simple. Let \( d(x) \) denote the plane tangent to \( h(x) \) at the point \( E\{x\} \). Then

\[
E\{h(x)\} \leq E\{d(x)\} = d(E\{x\}) = h(E\{x\})
\]

(A.11)

which proves (A.10). The inequality in (A.11) follows from the concavity of \( h(x) \), the first equality follows from the fact that \( d(x) \) is a linear function of \( x \), and the second equality from the fact that \( d(x) \) is tangent (and hence equal) to \( h(x) \) at the point \( E\{x\} \).

After these preparations, we turn our attention to the main question of finding a majorizing function for (A.9). Let \( z \) be a random vector whose
probability density function conditioned on \( y \) is completely determined by \( \theta \), and let

\[
g_i(\theta) = f(\theta^i) - E_{z|y, \theta^i} \left\{ \ln \left( \frac{p(y, z|\theta)}{p(y, z|\theta^i)} \right) \right\}.
\]  

(A.12)

Clearly \( g_i(\theta) \) satisfies:

\[
g_i(\theta^i) = f(\theta^i).
\]  

(A.13)

Furthermore, it follows from Jensen’s inequality (A.10), the concavity of the function \( \ln(\cdot) \), Bayes’ rule for conditional probabilities, and (A.9) that:

\[
g_i(\theta) \geq f(\theta^i) - \ln \left( \frac{1}{p(y|\theta^i)} \int_{p(y|\theta^i)} p(y, z|\theta) \, dz \right) \]

\[
= -\ln p(y|\theta^i) - \ln \left( \frac{p(y|\theta)}{p(y|\theta^i)} \right) = -\ln p(y|\theta) = f(\theta)
\]  

(A.14)

which shows that the function \( g_i(\theta) \) in (A.12) also satisfies the key majorization condition (A.5). Usually, \( z \) is called the unobserved data (to distinguish it from the observed data vector \( y \)), and the combination \((z, y)\) is called the complete data (sometimes \( y \) is called the incomplete data).

It follows from (A.13) and (A.14) along with the discussion in the previous section about the majorization approach that the following algorithm will monotonically reduce the negative log-likelihood function at each iteration:

\[
\theta^0 = \text{given}
\]

For \( i = 0, 1, 2, \ldots \) do:

**Expectation step**: Evaluate \( E_{z|y, \theta^i} \{ \ln p(y, z|\theta) \} \triangleq \mathcal{J}_i(\theta) \)

**Maximization step**: Compute \( \theta^{i+1} = \arg \max_{\theta} \mathcal{J}_i(\theta) \).

(A.15)

This is the EM algorithm in a nutshell.

An important aspect of the EM algorithm, which must be considered in every application, is the choice of the unobserved data vector \( z \). This choice should be done such that the maximization step of (A.15) is “easy” or, in any case, much easier than the maximization of the likelihood function. In general, doing so is not an easy task. In addition, the evaluation of the conditional expectation in (A.15) may also be rather challenging. We may speculate that somewhat paradoxically, the aforementioned difficulties associated with the use of the EM algorithm may have been the cause for its considerable popularity. Indeed, the detailed derivation of the EM algorithm for a particular problem is a more challenging research problem (and hence a more appealing one to many researchers) than, for instance, the derivation of a cyclic minimizer (which, when applicable, also possesses the key property (A.1) of the EM algorithm).
Appendix B

Proof of (3.22) (Chapter 3)

Let \( f(n) = V(\hat{\theta}_n) \) and \( g(n) = \frac{2}{N} \sum_{p=1}^{k} w_p^T \tilde{R}^{-1} w_p \). Then, by the definition of \( \hat{n}_{\text{STV}} \) and \( \hat{n}_{\text{LTV}} \) (see (3.16) and (3.17)) we have that:

\[
\begin{align*}
\left\{ \begin{array}{l} 
    f(\hat{n}_{\text{STV}}) + g(\hat{n}_{\text{STV}}) \leq f(\hat{n}_{\text{LTV}}) + g(\hat{n}_{\text{LTV}}) \\
    f(\hat{n}_{\text{LTV}}) + \mu g(\hat{n}_{\text{LTV}}) \leq f(\hat{n}_{\text{STV}}) + \mu g(\hat{n}_{\text{STV}})
\end{array} \right. \\
\end{align*}
\]

(B.1)

where \( \mu = k/2 > 1 \) (for \( N^c \geq e^3 \)). It follows from (B.1) that

\[
\begin{align*}
    f(\hat{n}_{\text{LTV}}) - f(\hat{n}_{\text{STV}}) \leq \mu [g(\hat{n}_{\text{STV}}) - g(\hat{n}_{\text{LTV}})] \leq \mu [f(\hat{n}_{\text{LTV}}) - f(\hat{n}_{\text{STV}})] \\
\end{align*}
\]

(B.2)

and hence that

\[
\begin{align*}
    f(\hat{n}_{\text{LTV}}) - f(\hat{n}_{\text{STV}}) \leq \mu [f(\hat{n}_{\text{LTV}}) - f(\hat{n}_{\text{STV}})]. \\
\end{align*}
\]

(B.3)

Because \( \mu > 1 \) the inequality in (B.3) is possible if and only if

\[
\begin{align*}
    f(\hat{n}_{\text{LTV}}) \geq f(\hat{n}_{\text{STV}}) \\
\end{align*}
\]

(B.4)

which in turn implies (3.22), as \( f(n) \) is monotonically decreasing with increasing \( n \) (under the made assumption that the model set has a nested structure).
Appendix C

Minimum Error Probability Threshold (Chapter 11)

In this appendix we describe how to obtain the threshold value that gives the minimum probability of making an error when deciding whether an element in the range profile corresponds to a target or clutter. Most of the theory used herein is described in Chapter 2.2 of [160]. We assume below that the true range profile $x$ is available, so the optimality condition will not hold when using the threshold with the estimated range profiles. However, without this assumption the detection becomes significantly more complex.\(^1\) We make the same assumptions on $x$ as we did for the semi-sparse MMSE/MAP estimate (see Section 11.3), i.e., that the elements in $x$ are independent and normally distributed, $x(j) \sim \mathcal{CN}(0, \gamma^2_j)$. The assumption that the elements in $x$ are independent allows us to perform the detection for each element separately. Below, we skip the sub index $j$ for notational convenience.

We have the following hypotheses for the value of $\gamma^2$:

$$
H_0 : \quad \gamma^2 = \nu^2 \quad \text{with a priori probability } P(H_0) = p \\
H_1 : \quad \gamma^2 = \rho^2 \quad \text{with a priori probability } P(H_1) = 1 - p
$$

(C.1)

where $H_0$ is the “clutter” hypothesis and $H_1$ is the “target” hypothesis.

It is easy to show that the magnitude of $x$, $z := |x|$, is a sufficient statistic for choosing between $H_0$ or $H_1$ [160]. The magnitude of $x$ can be shown to be Rayleigh distributed [142]:

$$p(z|\gamma^2) = \frac{2z}{\gamma^2} e^{-z^2/\gamma^2}.$$  

\(^1\)Specifically, the elements in $x$ are not independent and can therefore not be thresholded separately.
The total probability of error in our detector is defined as [160]

\[
Pr(\epsilon) = P(H_0)P_F + P(H_1)P_M
\]  

where \(P_F\) and \(P_M\) stands for the probability of false alarm (finding a target when we have clutter) and the probability of a miss (finding clutter when we have a target), respectively. If we use the threshold \(\tau\) and say that \(H_1\) is true when \(z > \tau\), and that \(H_0\) is true otherwise, we get the following expressions for \(P_F\) and \(P_M\):

\[
P_F = \int_{\tau}^{\infty} p(z|\nu^2)dz = \int_{\tau}^{\infty} \frac{2z}{\nu^2} e^{-z^2/\nu^2} dz = e^{-\tau^2/\nu^2}
\]

\[
P_M = \int_{0}^{\tau} p(z|\rho^2)dz = \int_{0}^{\tau} \frac{2z}{\rho^2} e^{-z^2/\rho^2} dz = 1 - e^{-\tau^2/\rho^2}.
\]

By inserting the above expressions and (C.1) in (C.2) we can minimize \(Pr(\epsilon)\) with respect to the threshold value \(\tau\):

\[
\hat{\tau} = \arg \min_{\tau} Pr(\epsilon) = \arg \min_{\tau} p \left( e^{-\tau^2/\nu^2} \right) + (1 - p) \left( 1 - e^{-\tau^2/\rho^2} \right)
\]

which is the value of \(\tau\) that gives the lowest probability of making a classification error under the assumptions made.
Bibliography


[133] ——, “RAKE receiver for channels with a sparse impulse response,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 9, pp. 3175–3180, September 2007.


