

tobira K

次世代信号情報処理シリーズ 6

Next SIP series

通信の信号処理

— 線形逆問題, 圧縮センシング,
確率推論, ウィルティンガー微分 —

田中聰久 監修
林 和則 著

コロナ社

tobira K■■■

(とびら窓)
白

シリーズ刊行のことば

信号処理とは、音声、音響、画像、電波など、連続する数値や連続波形が意味を持つデータを加工する技術です。現代の ICT 社会・スマート社会は信号処理なしには成り立ちません。スマートフォンやタブレットなどの情報端末はコンピュータ技術と信号処理技術が見事に融合した例ですが、私たちがその存在を意識することがないほど、身の回りに浸透しています。さらには、応用数学や最適化、また統計学を基礎とする機械学習などのさまざまな分野と融合しながらさらに発展しつつあります。

もともと信号処理は回路理論から派生した電気電子工学の一分野でした。抵抗、コンデンサ、コイルを組み合わせると、特定の周波数成分を抑制できるアナログフィルタを構成できます。アナログフィルタ技術は電子回路と融合することで能動フィルタを生み出しました。そしてデジタル回路の発明とともに、フィルタもデジタル化されました。一度サンプリングすれば、任意のフィルタをソフトウェアで構成できるようになったのです。ここに「デジタル信号処理」が誕生しました。そして、高速フーリエ変換の発明によって、デジタル信号処理は加速度的に発展・普及することになったのです。

デジタル技術によって、信号処理は単なる電気電子工学の一分野ではなく、さまざまな工学・科学と融合する境界分野に成長し始めました。フィルタのソフトウェア化は、環境やデータに柔軟に適応できる適応フィルタを生み出しました。信号はバッファリングできるようになり、画像信号はバッチ処理が可能になりました。そして、線形代数や統計学を柔軟に応用することで、テレビやカメラに革命をもたらしました。もともと周波数解析を基とする音声処理技術は、ビッグデータをいち早く取り込み、人工知能の基盤技術となっています。電波伝送の一分野だった通信工学は、通信のデジタル化によって信号処理技術

ii シリーズ刊行のことば

なしには成り立たないうえ、現代のスマート社会を支えるインフラとなっています。このように、枚挙に暇がないほど、信号処理技術は社会における各方面での基盤となっているだけでなく、さまざまな周辺技術と柔軟に融合し新たなテクノロジーを生み出しつつあります。

また、現代テクノロジーのコアたる信号処理は、電気・電子・情報系における大学カリキュラムでは必要不可欠な科目となっています。しかしながら、大学における信号処理教育はデジタルフィルタの設計に留まり、高度に深化した現代信号処理からはほど遠い内容となっています。一方で、最新の信号処理技術、またその周辺技術を知るには、論文を読んだり、洋書にあたったりする必要があります。さらに、高度に抽象化した現代信号処理は、ときに高等数学をバックグラウンドにしており、技術者は難解な数学を学ぶ必要があります。以上のことが本分野へ参入する壁を高くしているといえましょう。

これがまさに、次世代信号情報処理シリーズ“Next SIP”を刊行するに至ったきっかけです。本シリーズは、従来の伝統的な信号処理の専門書と、先端技術に必要な専門知識の間のギャップを埋めることを目的とし、信号処理分野で先端を走る若手・中堅研究者を執筆陣に揃えています。本シリーズによって、より多くの学生・技術者に信号処理の面白みが伝わり、さらには日本から世界を変えるイノベーションが生まれる助けになれば望外の喜びです。

2019年6月

次世代信号情報処理シリーズ監修 田中聰久

まえがき

現代の無線通信システムはますます大規模化・複雑化・高度化し、通信分野の初学者がその全容を理解しようとしたときにどこから手をつけてよいかすらわからない状況にあります。一方、現代の無線通信システムは、さまざまな信号処理技術がそのほぼすべての構成要素に埋め込まれることで成り立っています。デジタルコヒーレント方式が採用された近年の光通信システムも同様です。このため、通信システムで利用されている基本的な信号処理手法を理解することが、通信システム全体を理解するための近道となると考えられます。しかし、通信技術者はシステムをレイヤ（層）ごとに切り分けて設計することでその複雑さに対処してきたという歴史的な経緯もあり、通信に関する多くの教科書では、通信路等化や通信路応答推定、ビームフォーミング、ダイバーシティ合成、マルチユーザ検出、MIMO（multiple-input multiple-output）信号検出、無線資源割り当て、I/Q（in-phase/quadrature-phase）不完全補償、波長分散補償、偏波モード分散補償、非線形ひずみ補償、誤り訂正符号の復号など、要素技術ごとに信号処理手法が説明されており、通信システムで利用されている信号処理手法に対する統一的な理解を得ることが難しいという問題があります。

本書は通信分野の非専門家や大学生、大学院生などの初学者を対象としたもので、その特徴は、通信の典型的な問題に対する典型的な信号処理手法について解説することで、現在の通信システムで用いられている最先端の信号処理技術を理解するために必要な最低限の知識を、効率的かつ体系的に提供することを目的としていることです。

第1章では、準備として本書で使用する記号の定義と基礎事項について説明します。多くの通信システムでは正弦波で表現される搬送波の二つの直交する成分（同相成分と直交成分と呼ばれ、それぞれ cos 成分と sin 成分に相当しま

す)に情報を載せて送信しますが、オイラーの公式を利用して同相成分と直交成分をそれぞれ複素数の実部と虚部に対応させることにより、等価低域系(ベースバンド系)と呼ばれる複素数を利用した簡潔な入出力表現を得ることができます。通信の信号処理は、ほとんどの場合、ベースバンド系で考えるので、ここで複素数の基礎事項について復習します。また、信号処理ではさまざまな場面でベクトルの「大きさ(長さ)」を評価する必要がありますが、そのためにノルムと呼ばれる関数が利用されます。古典的には ℓ_2 ノルムやユークリッドノルムと呼ばれる、ベクトルの各成分の2乗和の平方根がよく利用されましたが、圧縮センシングなどの正則化を利用した最近の信号処理ではさまざまな種類のノルムが利用されるため、ここでノルムの基礎事項についてもおさらいします。

第2章では、確率変数や確率過程について復習します。特に通信の信号処理では、確率変数に時間インデックスを付けて並べたもの、すなわち確率過程として信号をモデル化することが多いので、確率過程やその相關行列の性質についてよく理解する必要があります。また、観測に基づいて何らかの推定を行うことは信号処理の最も基本的な問題の一つですが、そこでは、通常の確率変数の独立性だけでなく、ある確率変数に関する観測(実現値)が得られたという条件の下での独立性(条件付き独立性)が重要や役割を果たします。グラフィカルモデルは、多数の確率変数の中に存在する条件付き独立性を抽出し、それを利用した推論アルゴリズムを構成する際に強力なツールとなります。

第3章では、複素関数の微分について説明します。通信の信号処理では、複素数の信号に対して複素数の重み係数を用いてさまざまな処理を行います。複素数の重み係数を決める際には何らかの尺度でよいものを選択する必要がありますが、複素数は素朴に大小関係が決まらないため、複素数の重み係数を引数とし実数値の出力をもつ関数(コスト関数や目的関数)についての最適化問題を考えることになります。ところが、複素数の引数をもつ実数値関数は一般に正則(複素微分可能)ではなく、工学部の標準的なカリキュラムで学ぶ複素関数論の知識では最適なフィルタ係数を決定するのが難しいという問題があります。第3章で説明するウィルティンガー微分は、非正則な複素関数の勾配を効

ま　え　が　き

率的に計算することができる大変有用な手法です。このような数学的基礎に関する内容は既存の教科書では付録で説明されるのが普通ですが、 ウィルティンガー微分は通信の信号処理を理解する上で必要不可欠であり、 これについて説明した和書があまりないことから、 第3章で説明することにしました。どうしても微積分が苦手な方は、 この章の最後に「これだけ覚えておけばOK」という計算ルールをまとめていますので、 そこだけ目を通してつぎの章に進んでいただいても結構です。

第4章では、 通信の信号処理の典型的な問題設定として、 線形観測モデルに対する逆問題、 すなわち線形逆問題について考えます。具体的には、 興味のある未知ベクトル x に対して既知の行列 A で線形観測を行ったときに、 白色の加法性雑音 v を伴って観測されるベクトル $y = Ax + v$ から未知ベクトル x を推定する問題です。これは素朴ですが、 通信の多くの信号処理の問題を表現することが可能なモデルで、 極論すると通信の技術者はこのモデルで表現される問題しか考えていないといつてもよいほどです。実際、 上述の通信路等化や通信路応答推定などのほとんどの通信の問題が線形観測モデルで記述されます。ZF (zero-forcing) 推定や最小平均2乗誤差 (minimum mean-square-error, MMSE) 推定、 減算型干渉除去、 最大比合成、 最大事後確率推定、 最尤推定といった基本的な線形逆問題のための手法について説明し、 最後に通信応用の例にも触れます。

第5章でも線形観測モデルに基づく信号推定法について考えます。ただし、 第4章とは問題設定が異なり、 観測ベクトル y の次元が未知ベクトル x の次元よりも小さいことを想定します。これは劣決定線形観測モデルと呼ばれ、 未知ベクトルに関する何らかの先駆的な知識がない場合には、 観測ベクトル y から実際に観測された x を推定することが困難です。このような問題に対しては、 実際に観測された y に近いものが得られるような x のうち、 ノルムが小さいものを推定値として選ぶという方法、 すなわち正則化が利用されます。 x のノルムにはさまざまなものが考えられるため、 それに応じてさまざまな正則化法が存在します。従来は x の大きさを ℓ_2 ノルムで測る正則化法、 すなわち最小ノ

ルム解が広く利用されてきましたが、必ずしも実際に観測された x に近い推定値が得られるという保証はありません。一方、近年では、未知ベクトル x がスパースであること（成分のほとんどが 0 であること）があらかじめわかっているときに ℓ_1 ノルムによる正則化を用いると、ある条件の下で劣決定の線形観測 y から実際に観測された真の x が完全再構成できることが明らかにされたため、 ℓ_1 ノルムも正則化によく利用されます。 ℓ_1 正則化やそれを基にした手法は圧縮センシングやスパースモデリングと呼ばれ、いまでは信号処理や機械学習に必要不可欠な手法になっています。スパースベクトルの再構成アルゴリズムや再構成の条件についても第 5 章で説明します。

第 6 章では、ある空間に配置された複数のセンサやアンテナで得られた信号に対する信号処理、すなわちアレー信号処理について説明します。第 4 章で説明した線形観測モデルに基づく手法は、アンテナの指向性を制御するビームフォーミングなど、アレー信号処理においても有効ですが、ここでは最も基本的なアレー信号処理の一つである到来方向推定でよく利用される部分空間法について説明します。部分空間法は、センサアレーで観測された信号の相関行列の、異なる固有値に対応する固有ベクトルが直交する性質に基づく信号処理手法で、これを用いた MUSIC (multiple signal classification) 法は高分解能の到来方向推定が可能などで知られています。また、部分空間法と電波干渉計の原理に基づいた ESPRIT (estimation of signal parameters via rotational invariance techniques) 法も到来方向推定によく利用されます。到来方向推定は古典的な信号処理技術ですが、観測信号に擬定常性という性質を仮定すると、相関行列をベクトル化することで得られた信号がセンサアレーでの観測信号と同様の構造をもつことを利用して、実際のセンサ素子数よりも多くの到来波の到来方向を推定可能であることが明らかになるなど、最近大きな進展がありました。これらの新しいアプローチについても第 6 章で説明します。

第 7 章では、状態空間モデルに基づく推定法について説明します。状態空間モデルは制御分野でよく利用されますが、通信分野の専門家にはあまり馴染みがないかもしれません。線形ガウス型の状態空間モデルにおける観測モデルは

第4章で登場する線形観測モデルそのものですが、システムモデルは通信システムの議論で見かけることはあまりないので、通信分野の専門家にとってやや受け入れ難い印象があるようです。実は、状態空間モデルは、システムの状態を表す未知の確率変数と観測可能な確率変数の同時分布の分解を考え、状態変数と観測のそれぞれについてマルコフ性を仮定することで自然に導出されるモデルです。第7章では、状態空間モデルの導出から始めて、予測分布、フィルタ分布、平滑化分布の逐次的な計算法について説明します。粒子フィルタやカルマンフィルタによる具体的な状態推定法についても説明します。

第8章では、ベイズの定理に基づく確率推論問題について考えます。ベイズの定理自体は単なる条件付き分布の変換の式ですが、式中の確率変数に対して「興味のある未知確率変数」と「観測可能な確率変数」という役割を与えると、観測結果に基づいて未知変数を推論するための手順を与えてくれるきわめて有用な式となります。その際、観測可能な確率変数の実現値が与えられた下での、興味のある未知確率変数のそれぞれの条件付き確率分布（周辺事後分布）を求めることが主要な課題となります。実際、通信の信号検出の問題ではビット誤り率を最小にすることが求められますが、そのような信号検出は、周辺事後分布の最大値に対応する確率変数の値を推定値とする最大周辺事後確率推定によって実現されます。周辺事後分布の計算は素朴に行うと指数オーダーの計算量になるため、多変量の確率変数を扱う際には計算量的に破綻してしまいます。そこで、観測変数と未知変数の同時分布の因数分解を利用して効率的に周辺事後分布を計算するためのアルゴリズムが、第8章で説明する確率伝播法（belief propagation, BP）です。確率伝播法は第2章で説明するグラフィカルモデル上でメッセージを伝播する形で実行されるアルゴリズムで、シャノン限界に迫る符号として知られているターボ符号や低密度パリティ検査（low density parity check, LDPC）符号の復号、大規模MIMO信号検出や圧縮センシングにおけるスパース再構成アルゴリズムなどに応用されています。第7章の状態推定で登場する平滑化分布は、実は周辺事後分布そのもので、カルマンフィルタによる平滑化分布の計算は確率伝播法の特別な場合と考えられます。また、隠れマ

ルコフモデルに対する前向き後ろ向き (Bahl–Cocke–Jelinek–Raviv, BCJR) アルゴリズムや、さらには高速フーリエ変換までもが確率伝播法の原理によって導出されます。第8章では、ファクターグラフ上の sum-product アルゴリズムとペイジアンネットワーク上での Pearl の BP アルゴリズムについて説明し、確率伝播法の基本的な考え方とその原理を解説します。

本書は読者の皆さんのが独学できることを重視し、式変形などをできるだけ省略しないように説明しています。このため、同分野の他書に比べて容易に読み進められると思いますが、式変形がフォローできることと、その手法を理解していることには大きな隔たりがあります。筆者の恩師の原晋介先生にいただいた言葉の一つに「簡単に「わかった」と思ってはいけない」があります。本當は十分に理解できていないのに「わかった」と思い込んでしまうと、それ以降そのことについて学ぶ機会を失ってしまう危険性を指摘された言葉だと理解しています。本書で学ばれる際には白紙とペンを用意して、節ごとに学ばれた手法を何も見ないでそこに説明できるか確認してみてください。もし手が止まってしまったら、それはその節に書かれていることが十分に理解できていないということです。このような姿勢で本書を通読いただけたら、通信分野の研究を行うために必要な基礎体力が身につくことを請け合います。

本書の執筆の機会を与えていただき、初稿について多くの有益なコメントをいただきました「次世代信号情報処理シリーズ」監修の東京農工大学 田中聰久先生に厚く御礼申し上げます。また、名古屋工業大学 和田山正先生、大阪公立大学 大野修一先生、電気通信大学 石橋功至先生、大阪大学 早川諒先生をはじめ、多くの方から有益なコメントをいただきました。どうもありがとうございました。そして、本書の完成まで粘り強くお付き合いくださいましたコロナ社に心より感謝いたします。

2023年9月

林 和則

目 次

1. 記号と準備

1.1 記号の定義	1
1.2 複素数	2
1.3 ノルム	6
1.4 むすび	7

2. 確率変数と確率過程

2.1 確率の基本法則	8
2.2 期待値	13
2.3 確率過程	19
2.4 相関行列の性質	22
2.5 条件付き独立	29
2.6 グラフィカルモデル	31
2.6.1 ベイジアンネットワーク	31
2.6.2 ファクターグラフ	36
2.7 むすび	38

3. ウィルティンガー微分

3.1 実関数の微分	39
------------------	----

x 目 次

3.1.1 実 1 変数関数の微分	39
3.1.2 実関数の偏微分	40
3.1.3 実関数の全微分	41
3.1.4 全微分による勾配の計算	43
3.2 複素関数の微分	45
3.2.1 正則関数	45
3.2.2 正則でない複素関数の例	47
3.2.3 ウィルティンガー微分（スカラー引数）	48
3.2.4 ウィルティンガー微分（ベクトル引数）	50
3.2.5 複素勾配	52
3.2.6 ウィルティンガー微分の具体例	53
3.3 むすび	55

4. 線形逆問題のための基本的な手法

4.1 線形観測モデル	56
4.2 ZF 推定と最小 2 乗推定	58
4.2.1 ZF 推定	58
4.2.2 最小 2 乗推定	61
4.2.3 雑音強調	62
4.3 MMSE 推定	63
4.3.1 一般の MMSE 推定	64
4.3.2 線形 MMSE 推定	66
4.3.3 線形 MMSE 推定の SINR	68
4.4 減算型干渉除去	71
4.4.1 逐次干渉除去	72
4.4.2 並列干渉除去	74
4.5 信号合成	75

目 次 xi

4.5.1 選択合成	75
4.5.2 等利得合成	76
4.5.3 最大比合成	77
4.6 最大事後確率推定と最尤推定	80
4.7 通信応用の例	82
4.7.1 通信路等化	82
4.7.2 通信路推定	92
4.7.3 MIMO 信号検出	94
4.8 むすび	95

5. 圧縮センシング

5.1 最小ノルム解	96
5.2 スパース信号	98
5.3 圧縮センシングの考え方	100
5.4 再構成のアルゴリズム	103
5.5 再構成の条件	110
5.6 むすび	118

6. 部分空間法

6.1 アレー信号処理の基礎	119
6.2 信号部分空間と雑音部分空間	123
6.3 主成分分析とマイナー成分分析	127
6.4 MUSIC 法	130
6.5 空間平滑化	134
6.6 ESPRIT 法	139
6.7 KR 積拡張アレー処理による到来方向推定	143

6.7.1 KR 積とクロネッカー積, ベクトル化	144
6.7.2 KR 積拡張アレー処理	145
6.7.3 KR-MUSIC 法	147
6.8 む す び	150

7. 状 態 推 定

7.1 状態空間モデル	151
7.2 予測分布, フィルタ分布, 平滑化分布	154
7.2.1 予測分布	156
7.2.2 フィルタ分布	156
7.2.3 平滑化分布	157
7.3 粒子フィルタ	158
7.3.1 予測分布の計算	159
7.3.2 フィルタ分布の計算	161
7.3.3 SIR (sampling/importance resampling)	162
7.4 カルマンフィルタ	164
7.5 む す び	166

8. 確 率 推 論

8.1 確率推論問題	167
8.2 確率伝播法	170
8.2.1 確率伝播法の原理	170
8.2.2 sum-product アルゴリズム	173
8.2.3 Pearl の BP アルゴリズム	181
8.3 確率伝播法の応用	184
8.3.1 低密度パリティ検査 (LDPC) 符号	184

8.3.2 ターボ符号.....	187
8.3.3 高速フーリエ変換(FFT)	196
8.4 むすび	198

付録：よく使う行列に関する命題と性質

A.1 逆行列補題.....	199
A.2 クロネッカー積.....	201
A.3 ゲルシュゴリンの定理	203
引用・参考文献	205
索引	213

(p. xiv)
白

1

記号と準備

Next SIP

本章では準備として本書で使用する記号を定義し、通信の信号処理で必要となる複素数の基礎事項と、ベクトルの大きさを表すノルムについて説明します。

1.1 記号の定義

\mathbb{R} と \mathbb{C} はそれぞれ実数 (real number) と複素数 (complex number) の集合 (set) を表します。 \mathbb{R}^N と \mathbb{C}^N はそれぞれ成分が実数と複素数で次元が N のベクトルの集合を表します。 $\mathbb{R}^{M \times N}$ と $\mathbb{C}^{M \times N}$ はそれぞれ成分が実数と複素数でサイズが $M \times N$ の行列の集合を表します。ベクトルはポールド体の小文字で、行列はポールド体の大文字でそれぞれ表すものとし、別途定義しない限り、単にベクトルというときは列ベクトル (column vector) とします。サイズが $N \times N$ の単位行列は \mathbf{I}_N 、サイズが $M \times N$ で成分がすべて 0 である行列 (零行列 (zero matrix)) は $\mathbf{0}_{M \times N}$ でそれぞれ表し、文脈からサイズが明らかな場合には特にサイズを明記することもあります。 $\text{diag}[a_1 \ a_2 \ \cdots \ a_N]$ は a_1, a_2, \dots, a_N をこの順で対角成分にもつ $N \times N$ の対角行列 (diagonal matrix) を表します。 $\text{tr}\{\mathbf{A}\}$ は正方行列 \mathbf{A} のトレース (trace)、すなわち、対角成分の和を表します。 $\det\{\mathbf{A}\}$ は行列 \mathbf{A} の行列式 (determinant) を表します。

確率変数 (random variable) は大文字で、その実現値 (realization) は小文字で表記します[†]。すなわち、確率変数 X の実現値は x と書きます。ただし、

[†] 本書では、確率変数とは「確率的に値がばらつく変数」という程度の理解で差し支えありません。

2 1. 記号と準備

本書では行列とベクトルの表記も大文字と小文字で使い分けるので、誤解がないと考えられるときには確率変数と実現値同じ文字で表すことがあります。その場合には、確率変数なのか実現値なのかを文脈から判断するようにしてください。

1.2 複素数

通信の信号処理では、ほとんどの場合、扱う信号を複素数を用いて表現するため、複素数の基礎について理解しておくことが重要です。

複素数とは

$$j^2 = -1 \quad (1.1)$$

を満たす虚数単位 (imaginary unit) j と実数を組み合わせて作った数のことです[†]。複素数 $z \in \mathbb{C}$ は、一般に、実数 $x, y \in \mathbb{R}$ を用いて

$$z = x + jy \quad (1.2)$$

なる形で表されます。式 (1.2)において、 x は z の実部 (real part), y は z の虚部 (imaginary part) と呼ばれ、それぞれ本書では $x = \Re\{z\}$, $y = \Im\{z\}$ と表記します。

複素数 $z = x + jy$ に対して、虚部の符号を反転した複素数

$$z^* = x - jy \quad (1.3)$$

を z の複素共役 (complex conjugate) と呼びます。定義より明らかに

$$(z^*)^* = z \quad (1.4)$$

が成り立ちます。また、 z が実数のとき (つまり、 $y = 0$ のとき) は $z = z^*$ となり、逆に $z = z^*$ が成り立つなら z は実数です。

[†] 通常、虚数単位には文字 i が用いられます。しかし、通信を含む電気工学に関連する分野では、文字 i は電流に用いるため、慣例で文字 j が虚数単位としてよく利用されます。

二つの複素数 $z_1 = x_1 + jy_1, z_2 = x_2 + jy_2, x_1, x_2, y_1, y_2 \in \mathbb{R}$ に対して

$$\begin{aligned}
 (z_1 + z_2)^* &= (x_1 + jy_1 + x_2 + jy_2)^* \\
 &= \{(x_1 + x_2) + j(y_1 + y_2)\}^* \\
 &= (x_1 + x_2) - j(y_1 + y_2) \\
 &= x_1 - jy_1 + x_2 - jy_2 \\
 &= z_1^* + z_2^*
 \end{aligned} \tag{1.5}$$

なる性質をもちます。また

$$\begin{aligned}
 (z_1 z_2)^* &= \{(x_1 + jy_1)(x_2 + jy_2)\}^* \\
 &= \{(x_1 x_2 - y_1 y_2) + j(x_1 y_2 + x_2 y_1)\}^* \\
 &= (x_1 x_2 - y_1 y_2) - j(x_1 y_2 + x_2 y_1) \\
 &= x_1(x_2 - jy_2) - y_1(y_2 + jx_2) \\
 &= x_1(x_2 - jy_2) - jy_1(x_2 - jy_2) \\
 &= (x_1 - jy_1)(x_2 - jy_2) \\
 &= z_1^* z_2^*
 \end{aligned} \tag{1.6}$$

なる性質も信号処理でよく利用されます。

複素数は大きさや重さなどの数量を表す数ではないので、複素数に対して直接受けた不等式を考えることができません。そこで、複素数 $z = x + jy$ の大きさについて考えたい場合は

$$|z| = \sqrt{x^2 + y^2} \tag{1.7}$$

を考えます。これは複素数 z の**絶対値** (absolute value あるいは modulus) と呼ばれ、その定義から絶対値は非負の実数であることがわかります。

複素数 z の絶対値はその複素共役 z^* を用いて

$$|z| = \sqrt{zz^*} \tag{1.8}$$

4 1. 記号と準備

と書くこともできます。これを用いると、二つの複素数 $z_1 = x_1 + jy_1, z_2 = x_2 + jy_2, x_1, x_2, y_1, y_2 \in \mathbb{R}$ に対して

$$\begin{aligned} |z_1 z_2| &= \sqrt{(z_1 z_2)(z_1 z_2)^*} \\ &= \sqrt{z_1 \cdot z_2 \cdot z_1^* \cdot z_2^*} \\ &= \sqrt{z_1 z_1^*} \sqrt{z_2 z_2^*} \\ &= |z_1| |z_2| \end{aligned} \quad (1.9)$$

が成り立つことが示されます。

通信の信号処理では、 \cos や \sin を用いた信号の表現と複素数による信号の表現が、次のオイラーの公式によって結びつけられます。

$$e^{j\omega} = \cos \omega + j \sin \omega \quad (1.10)$$

ここで、 $\omega \in \mathbb{R}$ です。オイラーの公式を用いると

$$\cos \omega = \frac{e^{j\omega} + e^{-j\omega}}{2} \quad (1.11)$$

$$\sin \omega = \frac{e^{j\omega} - e^{-j\omega}}{2j} \quad (1.12)$$

が得られます。

$M \times N$ の実あるいは複素行列

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (1.13)$$

に対して、その転置 (transpose) A^\top は

$$A^\top = \begin{bmatrix} a_{0,0} & a_{1,0} & \cdots & a_{M-1,0} \\ a_{0,1} & a_{1,1} & \cdots & a_{M-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0,N-1} & a_{1,N-1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (1.14)$$

で定義されます。定義より、一般に

$$(A^T)^T = A \quad (1.15)$$

が成り立つことが確認できます。また、行列 A と B の積 AB が定義できるとき（つまり、 A の列数と B の行数が一致するとき）

$$(AB)^T = B^T A^T \quad (1.16)$$

が成り立つことも確認できます。一方

$$A^T = A \quad (1.17)$$

は一般に成り立ちませんが、これを満足する行列 A は**対称行列** (symmetric matrix) と呼ばれます。

行列 A が複素行列の場合は、その各成分の複素共役をとって転置させた行列

$$A^H = \begin{bmatrix} a_{0,0}^* & a_{1,0}^* & \cdots & a_{M-1,0}^* \\ a_{0,1}^* & a_{1,1}^* & \cdots & a_{M-1,1}^* \\ \vdots & \vdots & \ddots & \vdots \\ a_{0,N-1}^* & a_{1,N-1}^* & \cdots & a_{M-1,N-1}^* \end{bmatrix} \quad (1.18)$$

をよく考えます。これは行列 A の**エルミート転置** (Hermitian transpose), **共役転置** (conjugate transpose), **随伴行列** (adjoint matrix) と呼ばれます。転置と同様に、一般に

$$(A^H)^H = A \quad (1.19)$$

$$(AB)^H = B^H A^H \quad (1.20)$$

が成り立つことが確認できます。また

$$A^H = A \quad (1.21)$$

を満足する行列 A は**エルミート行列** (Hermitian matrix) と呼ばれます。

1.3 ノルム

ベクトルの大きさを表すのに用いられるのがノルム (norm) です。ノルムの定義にはさまざまなものがありますが、最もよく利用されるのが ℓ_2 ノルムで、実数あるいは複素数のベクトル $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^\top$ に対してつぎで定義されます。

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^N |x_i|^2} \quad (1.22)$$

これはユークリッドノルム (Euclidean norm) とも呼ばれ、我々が生活している3次元空間での通常の意味での距離をそのまま高次元空間に拡張したものです。多くの工学の問題で出てくるノルムはほとんどの場合 ℓ_2 ノルムであり、単にノルムといったときには ℓ_2 ノルムを指すといってよいほど代表的なものですが、一般にはつぎの公理を満足する関数 $\|\cdot\|$ は何でもノルムになります^{1)†1}。

定義 1.1 (ノルムの公理) K を \mathbb{R} または \mathbb{C} とし、 V を K 上のベクトル空間 (vector space)^{†2}とする。このとき任意の $\alpha \in K$ と任意の $\mathbf{x}, \mathbf{y} \in V$ に対して、つぎを満たす V から \mathbb{R} への関数 $\|\cdot\|$ をノルムという。

$$(i) \quad \|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0} \text{ (独立性)} \quad (1.23)$$

$$(ii) \quad \|\alpha \mathbf{x}\| = |\alpha| \|\mathbf{x}\| \text{ (齊次性)} \quad (1.24)$$

$$(iii) \quad \|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \text{ (劣加法性)} \quad (1.25)$$

直観的にはベクトルの各成分の絶対値の和、すなわち

$$\|\mathbf{x}\|_1 = \sum_{i=1}^N |x_i| \quad (1.26)$$

^{†1} 肩つき数字は巻末の引用・参考文献を示します。

^{†2} ベクトル空間の定義 (公理) については、文献2) や3) などを参照してください。

はベクトルの大きさを表していそうな気がしますが、実際これはノルムの公理を満足し、 ℓ_1 ノルムと呼ばれます。

より一般に、 $p \geq 1$ に対して

$$\|x\|_p = \left(\sum_{i=1}^N |x_i|^p \right)^{\frac{1}{p}} \quad (1.27)$$

はノルムであり、 ℓ_p ノルムと呼ばれます。 ℓ_2 ノルムや ℓ_1 ノルムは ℓ_p ノルムでそれぞれ $p = 2, 1$ とした特別な場合に相当することがすぐにわかります。

式 (1.27) を $0 < p < 1$ に対して用いると、ノルムの公理の 3 番目の式（三角不等式 (triangle inequality) とも呼ばれます）を満足しませんが、このような量は準ノルム (quasinorm) と呼ばれ、圧縮センシングなどの文脈でよく利用されます。さらに、つぎの準ノルムですらない量

$$\|x\|_0 = \text{card}(\text{supp}(x)) \quad (1.28)$$

を考えることがあります。ここで、 $\text{supp}(x)$ は x の台 (support) で $\text{supp}(x) = \{i : x_i \neq 0\}$ によって定義されます。また、 $\text{card}(\cdot)$ は集合の要素数を表します。したがって、 $\|x\|_0$ は x の非零要素数を意味し、慣例により ℓ_0 ノルムと呼ばれます。ただし、 ℓ_p ノルムで $p \rightarrow 0$ とした量とは一致しないので注意しましょう。

1.4 む す び

本章では、本書で使用する記号の定義を与えました。そして、第 2 章以降の内容を説明するための準備として、複素数の基礎事項とノルムについて簡単に説明しました。複素関数論についてより詳しく学びたい方には文献4) がお勧めです。本章では説明しませんでしたが、ノルムは行列に対しても定義されます。行列ノルムについては文献5) などを参照してください。

2

確率変数と確率過程

Next SIP

信号処理では、多くの場合、信号や雑音をインデックス（時刻など）付きの確率変数の集合、すなわち確率過程として扱います。また、通信の信号処理では複雑な通信路の影響を確率的な現象として取り扱うことが多いです。このため、確率変数や確率過程の基礎事項についてしっかりと理解しておくことはさまざまな信号処理手法をマスターするための必須の条件です。特に、相関行列の性質と条件付き独立という概念についてよく理解しましょう。

2.1 確率の基本法則

離散値をとる二つの確率変数 X, Y を考えます。ただし、 X は $x_i, i = 1, 2, \dots, M$ の値を、 Y は $y_j, j = 1, 2, \dots, N$ の値を、それぞれとるものとします。このとき、 X が x_i の値をとる確率 $\Pr(X = x_i)$ が、ある関数 $P_X(x)$ を用いて

$$\Pr(X = x_i) = P_X(x_i), \quad i = 1, 2, \dots, M \quad (2.1)$$

と書けるとき、 $P_X(x)$ のことを確率変数 X の確率分布（probability distribution）といいます。確率分布は

$$0 \leq P_X(x) \leq 1 \quad (2.2)$$

$$\sum_x P_X(x) = 1 \quad (2.3)$$

という性質をもちます。ただし、 \sum_x は確率変数 X のすべての可能な実現値 x について和をとることを意味します。

$X = x_i$ かつ $Y = y_j$ となる確率 $\Pr(X = x_i, Y = y_j)$ は、**同時確率**(simultaneous probability)や**結合確率**(joint probability)と呼ばれます。ある関数 $P_{XY}(x, y)$ を用いて

$$\Pr(X = x_i, Y = y_j) = P_{XY}(x_i, y_j), \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N \quad (2.4)$$

と書けるとき、 $P_{XY}(x, y)$ を確率変数 X, Y の**同時確率分布**(simultaneous probability distribution)あるいは**結合確率分布**(joint probability distribution)といいます。

$X = x_i$ という条件の下で $Y = y_j$ となる確率 $\Pr(Y = y_j | X = x_i)$ は、**条件付き確率**(conditional probability)と呼ばれます。ある関数 $P_{Y|X}(y|x)$ を用いて

$$\Pr(Y = y_j | X = x_i) = P_{Y|X}(y_j | x_i), \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N \quad (2.5)$$

と書けるとき、 $P_{Y|X}(y|x)$ を X が与えられた下での確率変数 Y の**条件付き確率分布**(conditional probability distribution)といいます。

以上の準備の下で、確率計算の最も基本的なルールは以下のとおりです⁶⁾。

定理 2.1 (離散確率変数の加法定理 (addition theorem))

$$P_X(x) = \sum_y P_{XY}(x, y) \quad (2.6)$$

定理 2.2 (離散確率変数の乗法定理 (multiplication theorem))

$$P_{XY}(x, y) = P_{Y|X}(y|x)P_X(x) \quad (2.7)$$

加法定理の式 (2.6) の操作は周辺化 (marginalization) とも呼ばれます。また、乗法定理の式 (2.7) は条件付き確率の定義にもなっています。

同時確率分布が

$$P_{XY}(x, y) = P_X(x)P_Y(y) \quad (2.8)$$

と書けるとき、確率変数 X と Y は独立 (independent) であるといい、 $X \perp Y$ と表記します。このとき、乗法定理より $P_{Y|X}(y|x) = P_Y(y)$ となり、 X が与えられたという条件が確率変数 Y に影響しないことを意味します。

3 変数以上の確率変数間の独立性を考えるときには、「独立」と「たがいに独立」の違いについて注意する必要があります†。例えば、確率変数 X, Y, Z が独立 (mutually independent) であるとは、同時分布が

$$P_{XYZ}(x, y, z) = P_X(x)P_Y(y)P_Z(z) \quad (2.9)$$

と書けることをいいます。一方、確率変数 X, Y, Z がたがいに独立 (pairwise independent) とは

$$P_{XY}(x, y) = P_X(x)P_Y(y) \quad (2.10)$$

$$P_{YZ}(y, z) = P_Y(y)P_Z(z) \quad (2.11)$$

$$P_{XZ}(x, z) = P_X(x)P_Z(z) \quad (2.12)$$

が成り立つことをいいます。独立な確率変数はたがいに独立ですが、たがいに独立な確率変数は必ずしも独立ではありません⁷⁾。

よく用いられる他の計算規則として、ベイズの定理 (Bayes' theorem) があります。同時確率の対称性の式、つまり

$$P_{XY}(x, y) = P_{YX}(y, x) \quad (2.13)$$

において、その両辺に乗法定理を用いることで

† 通常、英語の「mutually」は「たがいに」と和訳されますが、このことが誤解を招く原因になっているように思われます。

$$P_{Y|X}(y|x)P_X(x) = P_{X|Y}(x|y)P_Y(y) \quad (2.14)$$

となり、次式が得られます。

定理 2.3 (ベイズの定理)

$$P_{X|Y}(x|y) = \frac{P_{Y|X}(y|x)P_X(x)}{\sum_x P_{Y|X}(y|x)P_X(x)} \quad (2.15)$$

ベイズの定理は単に条件付き確率間の関係を表す式ですが、それぞれの確率変数に役割をもたせると、後述する確率推論 (probabilistic inference) において中心的な役割を果たす重要な意味をもつ式になります。

つぎに連続値をとる確率変数について考えましょう。実数値（あるいは複素数値）をとるスカラーの確率変数 X が任意の $c \in \mathbb{R}$ （複素数の場合は $c \in \mathbb{C}$ ）に対して $\Pr(X = c) = 0$ となるとき、 X は連続の（分布をもつ）確率変数であるといいます。

簡単のため実数値の場合を考えます。連続の確率変数 X に対してある関数 $p_X(x) \geq 0, -\infty < x < \infty$ が存在して、 $a \leq X \leq b$ となる事象の確率が

$$\Pr(a \leq X \leq b) = \int_a^b p_X(x)dx \quad (2.16)$$

と書けるとき、 $p_X(x)$ を確率変数 X の確率密度関数 (probability density function, PDF) といいます[†]。確率密度関数は

$$p_X(x) \geq 0 \quad (2.17)$$

$$\int_{-\infty}^{\infty} p_X(x)dx = 1 \quad (2.18)$$

なる性質をもちますが、 $p_X(x)$ 自体は確率ではないため、ある x に対して $p_X(x) > 1$ となり得ることに注意します。

[†] これに対して、離散確率変数の場合に考えた $P_X(x)$ を確率質量関数 (probability mass function, PMF) ということがあります。

確率変数 X が区間 $(-\infty, a)$ の値をとる確率を表す関数

$$c_X(a) = \Pr(X \leq a) \quad (2.19)$$

は累積分布関数 (cumulative distribution function, CDF) と呼ばれます。累積分布関数と確率密度関数には

$$c_X(a) = \int_{-\infty}^a p_X(x)dx \quad (2.20)$$

なる関係があります。なお、応用上はほとんど気にする必要はありませんが、理論的にはすべての確率分布が確率密度関数をもつわけではない（カントール分布⁸⁾など）ことに注意しましょう。

二つの連続確率変数 X, Y について考えます。1 変数の場合の議論と同様に、 X, Y の同時分布の確率密度関数 $p_{XY}(x, y)$ と条件付き分布の確率密度関数 $p_{Y|X}(y|x)$ を定義すると、離散確率変数の場合と同様に加法と乗法の計算ルールがつぎで与えられます⁹⁾。

定理 2.4 (連続確率変数の加法定理)

$$p_X(x) = \int p_{XY}(x, y)dy \quad (2.21)$$

定理 2.5 (連続確率変数の乗法定理)

$$p_{XY}(x, y) = p_{Y|X}(y|x)p_X(x) \quad (2.22)$$

連続確率変数の場合は、同時分布の密度関数に対して

$$p_{XY}(x, y) = p_X(x)p_Y(y) \quad (2.23)$$

が成り立つとき、 X, Y は独立であるといい、 $X \perp\!\!\!\perp Y$ と書きます。

$\delta(x)$ を、任意の連続関数 $f(x)$ に対して

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0) \quad (2.24)$$

なる性質を満たす関数（ディラックのデルタ関数）と定義すると、 $\delta(x)$ を用いることで確率密度関数と確率質量関数の重み付け和のような形で与えられる分布も表現できます（一般化された確率密度関数と呼びます）^{†1}。本書では、離散確率変数と連続確率変数を同時に考える必要がある場合には、特に明記することなく一般化された確率密度関数を仮定します。

2.2 期 待 値

統計的信号処理では、確率変数を引数とするさまざまな関数（一般に統計量（statistic）と呼ばれ、それ自身も確率変数となります）の平均を考えます。これは期待値（expected value）と呼ばれ、例えば、確率変数 X を引数にもつ関数 $f(X)$ の期待値は $E[f(X)]$ と書かれます。ここで、 $E[\cdot]$ は期待値をとる操作（expectation）を表します^{†2}。

具体的には、 X が確率密度関数 $p_X(x)$ をもつ連続確率変数の場合は

$$E[f(X)] = \int f(x)p_X(x)dx \quad (2.25)$$

と計算されます。一方、 X が確率分布 $P_X(x)$ をもつ離散確率変数の場合には

$$E[f(X)] = \sum_x f(x)P_X(x) \quad (2.26)$$

と計算されます。

関数 $f(X) = X$ の場合には、 $E[f(X)]$ は確率変数 X そのものの期待値

$$m_X = E[X] \quad (2.27)$$

^{†1} 式 (2.24) の性質を満足する関数は、通常の関数の範囲には存在しないことに注意しましょう。

^{†2} 文献によって $E[\cdot]$ など他の表記が用いられることもあります。

となります。

確率変数 X が複素数値をとるとき、関数

$$f(X) = |X - m_X|^2 = (X - m_X)(X - m_X)^* \quad (2.28)$$

の期待値

$$\sigma_X^2 = E[(X - m_X)(X - m_X)^*] \quad (2.29)$$

は確率変数 X の分散 (variance) と呼ばれます。

$E[\cdot]$ の演算は、具体的には引数の関数に確率密度関数を乗算して積分する（離散確率変数の場合には、確率分布を乗算して和をとる）演算なので、積分や和の演算が線形性 (linearity) をもつことから $E[\cdot]$ も線形の演算です。この性質を利用すると、確率変数 X の分散は

$$\begin{aligned} \sigma_X^2 &= E[|X|^2 - m_X X^* - X m_X^* + |m_X|^2] \\ &= E[|X|^2] - m_X E[X^*] - E[X] m_X^* + |m_X|^2 \\ &= E[|X|^2] - |m_X|^2 \end{aligned} \quad (2.30)$$

と書けることが示されます。ここで、 $m_X (= E[X])$ や $|m_X|^2$ は確定的な値をもつので、期待値操作 $E[\cdot]$ の外に出されることに注意します。

確率変数 X を a 倍してそれに b を足すことで得られる確率変数 $aX + b$ ($a, b \in \mathbb{C}$ は定数) の期待値は

$$m_{aX+b} = E[aX + b] = aE[X] + b \quad (2.31)$$

となり、元の確率変数 X の期待値を a 倍して b を足したものになります。また、確率変数 $aX + b$ の分散は

$$\begin{aligned} \sigma_{aX+b}^2 &= E[\{(aX + b) - E[aX + b]\}\{(aX + b) - E[aX + b]\}^*] \\ &= |a|^2 E[(X - E[X])(X - E[X])^*] \end{aligned} \quad (2.32)$$

となり、 b に依存せず、元の確率変数 X の分散を $|a|^2$ 倍したものになります。これらの性質は信号処理や統計で頻繁に利用されます。

つぎに二つの確率変数を引数にもつ関数の期待値を考えます。確率変数 X, Y が同時分布の確率密度関数 $p_{XY}(x, y)$ をもつ連続確率変数の場合、確率変数 X, Y を引数にもつ関数 $f(X, Y)$ の期待値 $E[f(X, Y)]$ は

$$E[f(X, Y)] = \int \int f(x, y) p_{XY}(x, y) dx dy \quad (2.33)$$

と計算されます。また、確率変数 X, Y が同時確率分布 $P_{XY}(x, y)$ をもつ離散確率変数の場合には

$$E[f(X, Y)] = \sum_x \sum_y f(x, y) P_{XY}(x, y) \quad (2.34)$$

と計算されます。

関数 $f(X, Y) = XY^*$ の場合には、 $f(X, Y)$ の期待値は

$$r_{X,Y} = E[XY^*] \quad (2.35)$$

となり、これを確率変数 X と Y の**相関** (correlation) と呼びます。特に

$$E[XY^*] = E[X]E[Y]^* \quad (2.36)$$

が成り立つとき、 X と Y は**無相関** (uncorrelated) といいます。 X と Y が無相関で、 $E[X]$ と $E[Y]$ のいずれかが 0 のとき、 X と Y の相関は 0 になります。

関数 $f(X, Y) = (X - E[X])(Y - E[Y])^*$ の場合には、 $f(X, Y)$ の期待値は

$$c_{X,Y} = E[(X - E[X])(Y - E[Y])^*] \quad (2.37)$$

となり、これを確率変数 X と Y の**共分散** (covariance) と呼びます。共分散は、相関を用いて

$$\begin{aligned} c_{X,Y} &= E[XY^* - E[X]Y^* - XE[Y]^* + E[X]E[Y]^*] \\ &= E[XY^*] - E[E[X]Y^*] - E[XE[Y]^*] + E[E[X]E[Y]^*] \end{aligned}$$

$$\begin{aligned}
 &= E[XY^*] - E[X]E[Y]^* - E[X]E[Y]^* + E[X]E[Y]^* \\
 &= E[XY^*] - E[X]E[Y]^* \tag{2.38}
 \end{aligned}$$

と表すことができるので、式(2.36)より X と Y が無相関のときは共分散 $c_{X,Y} = 0$ となります。

関数 $f(X, Y) = X + Y$ の場合には、 $f(X, Y)$ の期待値は一般に

$$m_{X+Y} = E[X] + E[Y] \tag{2.39}$$

となり、確率変数の和の期待値はそれぞれの確率変数の期待値の和となります。また、 $f(X, Y) = \{(X + Y) - E[X + Y]\}\{(X + Y) - E[X + Y]\}^*$ とするとその期待値は確率変数 $X + Y$ の分散であり

$$\begin{aligned}
 \sigma_{X+Y}^2 &= E[f(X, Y)] \\
 &= E[\{(X + Y) - E[X + Y]\}\{(X + Y) - E[X + Y]\}^*] \\
 &= E[\{(X - E[X]) + (Y - E[Y])\}\{(X - E[X]) + (Y - E[Y])\}^*] \\
 &= E[|X - E[X]|^2] + E[(X - E[X])(Y - E[Y])^*] \\
 &\quad + E[(X - E[X])^*(Y - E[Y])] + E[|Y - E[Y]|^2] \tag{2.40}
 \end{aligned}$$

となります。ここで、確率変数 X と Y が無相関、すなわち $E[(X - E[X])(Y - E[Y])^*] = 0$ とすると

$$\sigma_{X+Y}^2 = E[|X - E[X]|^2] + E[|Y - E[Y]|^2] \tag{2.41}$$

となることから、無相関の確率変数の和の分散は、それぞれの確率変数の分散の和になることがわかります。

なお、確率変数 X と Y が独立であるとき、これらは無相関になります。このことは、例えば、確率変数 X と Y が連続確率変数であるとすると

$$E[XY^*] = \int \int xy^* p_{XY}(x, y) dx dy$$

$$\begin{aligned}
 &= \int \int xy^* p_X(x) p_Y(y) dx dy \\
 &= \left(\int xp_X(x) dx \right) \left(\int yp_Y(y) dy \right)^* \\
 &= E[X]E[Y]^*
 \end{aligned}$$

となることからただちに確認できます。離散確率変数の場合も同様です。ただし、確率変数 X と Y が無相関だからといって、必ずしも独立とは限らないので注意しましょう[†]。

さらに、各成分が複素数値をとる N 変量の確率変数

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \tag{2.42}$$

について考えます。 \mathbf{X} が同時分布の確率密度関数 $p_{\mathbf{X}}(\mathbf{x})$ をもつ連続確率変数の場合、その期待値は

$$\mathbf{m}_{\mathbf{X}} = E[\mathbf{X}] = \int \mathbf{x} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \begin{bmatrix} E[X_1] \\ E[X_2] \\ \vdots \\ E[X_N] \end{bmatrix} \tag{2.43}$$

で与えられるベクトルです。 \mathbf{X} が離散確率変数の場合も、同時確率分布を用いて同様に定義されます。

N 変量の確率変数 \mathbf{X} について、**相関行列** (correlation matrix) は

$$\mathbf{R}_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^H]$$

[†] ガウス確率変数については、「無相関ならば独立」が成り立ちます。

$$\begin{aligned}
&= E \begin{bmatrix} X_1 X_1^* & X_1 X_2^* & \cdots & X_1 X_N^* \\ X_2 X_1^* & X_2 X_2^* & \cdots & X_2 X_N^* \\ \vdots & \vdots & \ddots & \vdots \\ X_N X_1^* & X_N X_2^* & \cdots & X_N X_N^* \end{bmatrix} \\
&= \begin{bmatrix} E[X_1 X_1^*] & E[X_1 X_2^*] & \cdots & E[X_1 X_N^*] \\ E[X_2 X_1^*] & E[X_2 X_2^*] & \cdots & E[X_2 X_N^*] \\ \vdots & \vdots & \ddots & \vdots \\ E[X_N X_1^*] & E[X_N X_2^*] & \cdots & E[X_N X_N^*] \end{bmatrix} \\
&= \begin{bmatrix} r_{X_1, X_1} & r_{X_1, X_2} & \cdots & r_{X_1, X_N} \\ r_{X_2, X_1} & r_{X_2, X_2} & \cdots & r_{X_2, X_N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{X_N, X_1} & r_{X_N, X_2} & \cdots & r_{X_N, X_N} \end{bmatrix} \tag{2.44}
\end{aligned}$$

と定義されます。

一方、共分散行列 (covariance matrix) は

$$\begin{aligned}
C_X &= E[(X - m_X)(X - m_X)^H] \\
&= \begin{bmatrix} E[(X_1 - m_{X_1})(X_1 - m_{X_1})^*] & E[(X_1 - m_{X_1})(X_2 - m_{X_2})^*] \\ E[(X_2 - m_{X_2})(X_1 - m_{X_1})^*] & E[(X_2 - m_{X_2})(X_2 - m_{X_2})^*] \\ \vdots & \vdots \\ E[(X_N - m_{X_N})(X_1 - m_{X_1})^*] & E[(X_N - m_{X_N})(X_2 - m_{X_2})^*] \\ \cdots & E[(X_1 - m_{X_1})(X_N - m_{X_N})^*] \\ \cdots & E[(X_2 - m_{X_2})(X_N - m_{X_N})^*] \\ \ddots & \vdots \\ \cdots & E[(X_N - m_{X_N})(X_N - m_{X_N})^*] \end{bmatrix}
\end{aligned}$$

$$= \begin{bmatrix} \sigma_{X_1}^2 & c_{X_1, X_2} & \cdots & c_{X_1, X_N} \\ c_{X_2, X_1} & \sigma_{X_2}^2 & \cdots & c_{X_2, X_N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{X_N, X_1} & c_{X_N, X_2} & \cdots & \sigma_{X_N}^2 \end{bmatrix} \quad (2.45)$$

で定義されます。 C_X はその要素に X の各成分の分散と共分散を含むので、**分散共分散行列** (variance-covariance matrix) と呼ぶこともあります。

2.3 確率過程

確率過程 (stochastic processあるいは random process) は、それぞれにインデックスが付与された確率変数の集合として定義されます。例えば、離散時間の時刻 n をインデックスとして用いると、各時刻の確率変数の集合 $\{X(n); n = \dots, -2, -1, 0, 1, 2, \dots\}$ は確率過程になります（**時系列** (time series) と呼びます）。ここで時刻 n をある特定の値 n_0 に固定すると、 $X(n_0)$ は確率変数となります。各時刻に対応する確率変数は離散と連続の両方の場合が考えられます。以下では、簡単のため任意の時刻 n に固定した $X(n)$ がすべて複素数値となる連続確率変数であるとします。

確率過程 $X(n)$ の期待値は一般に時刻 n に依存するため

$$m_X(n) = E[X(n)] \quad (2.46)$$

のように n を引数をもつ関数になります。一方、ある時刻 n とそこから k だけシフトした時刻 $n - k$ に対応する確率変数（の複素共役）の積の期待値

$$r_X(n, k) = E[X(n)X^*(n - k)] \quad (2.47)$$

を確率過程 $X(n)$ の**自己相関関数** (autocorrelation function) と呼びます。また、時刻 n と $n - k$ に対応する確率変数の共分散

$$c_X(n, k) = E[(X(n) - m_X(n))(X(n - k) - m_X(n - k))^*] \quad (2.48)$$

は確率過程 $X(n)$ の**自己共分散関数** (autocovariance function) といいます。

$k = 0$ のときの自己共分散関数

$$\sigma_X^2(n) = c_X(n, 0) \quad (2.49)$$

$$= E[(X(n) - m_X(n))(X(n) - m_X(n))^*] \quad (2.50)$$

は各時刻の分散を表し、分散の有界な確率過程を**2次過程** (second order process) といいます[†]。

式 (2.47) の自己相関関数や式 (2.48) の自己共分散関数は一般に二つの時刻 n と $n - k$ に依存するため、 n と k を引数にもちます。しかしながら、通信を含む多くの信号処理の問題では、信号の統計的な性質が時間的に変化しない（あるいは、観測時間内での変化が無視できる）ことが仮定できます。平均値が時刻 n に依存せず（つまり $m_X(n) = m_X$ と書け）、自己相関関数が時間差 k のみに依存する（つまり $r_X(n, k) = r_X(k)$ と書ける）2次過程は、**広義定常過程** (wide-sense stationary process) または**弱定常過程** (weakly stationary process) と呼ばれます。広義定常過程は、平均と自己相関関数が時刻の原点シフト $n \rightarrow n + l$, $\forall l$ に依存しない2次過程ということもできます。

広義定常よりも強い定常性の概念として**狭義定常**があります。確率過程 $X(n)$ の任意の有限次元分布の密度関数 $p_{X(n_1)X(n_2)\dots X(n_p)}(x(n_1), x(n_2), \dots, x(n_p))$ が時刻の原点シフト $n \rightarrow n + l$, $\forall l$ に対して不变のとき、つまり

$$\begin{aligned} & p_{X(n_1)X(n_2)\dots X(n_p)}(x(n_1), x(n_2), \dots, x(n_p)) \\ &= p_{X(n_1+l)X(n_2+l)\dots X(n_p+l)}(x(n_1 + l), x(n_2 + l), \dots, x(n_p + l)), \quad \forall l \end{aligned} \quad (2.51)$$

が成り立つとき、 $X(n)$ は**狭義定常過程** (strict-sense stationary process) または**強定常過程** (strongly stationary process) と呼ばれます。 $m_X(n)$ と $r_X(n, k)$ の定義に代入することで、狭義定常の確率過程は広義定常であることがただちに確認できます。ただし、その逆は一般に成り立ちません。

[†] 通常、我々が扱う信号の分散は有界ですが、コーシー分布や t 分布に従う確率変数など分散が有界でない確率変数もあります。

一方、定常でない確率過程は**非定常過程** (non-stationary process) と呼ばれます。統計的性質が時間的にでたらめに変化するものばかりではなく、統計的性質がある規則をもって時間変化するようなものもあります。その代表例が**周期定常過程** (cyclostationary process) で、信号処理でもしばしば登場します。周期定常性には通常の定常性と同様に広義と狭義の定義があります。確率過程 $X(n)$ は、任意の整数 m に対して

$$m_X(n) = m_X(n + mT) \quad (2.52)$$

$$r_X(n, k) = r_X(n + mT, k) \quad (2.53)$$

が成り立つとき、周期 T の広義周期定常過程と呼ばれます。本書では周期定常過程についてはほとんど扱いませんが、興味のある方は文献10)などを参照してください。

離散時間確率過程 $X(n)$ の複数の観測からなるベクトル

$$\mathbf{X}(n) = [X(n) \ X(n-1) \ \cdots \ X(n-N+1)]^\top \quad (2.54)$$

の相関行列は**自己相関行列** (autocorrelation matrix) と呼ばれ

$$\begin{aligned} \mathbf{R}_{\mathbf{X}(n)} &= E[\mathbf{X}(n)\mathbf{X}^H(n)] \\ &= \begin{bmatrix} E[X(n)X^*(n)] & \cdots & E[X(n)X^*(n-N+1)] \\ E[X(n-1)X^*(n)] & \cdots & E[X(n-1)X^*(n-N+1)] \\ \vdots & & \vdots \\ E[X(n-N+1)X^*(n)] & \cdots & E[X(n-N+1)X^*(n-N+1)] \end{bmatrix} \\ &= \begin{bmatrix} r_X(n, 0) & \cdots & r_X(n, N-1) \\ r_X(n-1, -1) & \cdots & r_X(n-1, N-2) \\ \vdots & & \vdots \\ r_X(n-N+1, -N+1) & \cdots & r_X(n-N+1, 0) \end{bmatrix} \quad (2.55) \end{aligned}$$

と定義されます。特に $X(n)$ が広義定常過程のときには

$$\mathbf{R}_{\mathbf{X}(n)} = \begin{bmatrix} r_X(0) & r_X(1) & \cdots & r_X(N-1) \\ r_X(-1) & r_X(0) & \cdots & r_X(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_X(-N+1) & r_X(-N+2) & \cdots & r_X(0) \end{bmatrix} \quad (2.56)$$

となり、左から右への各降下対角線上の成分が等しい値をもつテプリツツ行列 (Toeplitz matrix) と呼ばれる構造をもちます。ここで、主対角成分はすべて実数になることに注意します。

$\mathbf{X}(n)$ の共分散行列は自己共分散行列 (autocovariance matrix) と呼ばれ

$$\mathbf{C}_{\mathbf{X}(n)} = E[(\mathbf{X}(n) - E[\mathbf{X}(n)])(\mathbf{X}(n) - E[\mathbf{X}(n)])^H] \quad (2.57)$$

で定義されます。本書では別途定義しない限り $E[\mathbf{X}(n)] = \mathbf{0}$ と仮定しますが、このとき、自己相関行列 $\mathbf{R}_{\mathbf{X}(n)}$ と自己共分散行列 $\mathbf{C}_{\mathbf{X}(n)}$ は一致します。なお、 $X(n)$ が広義定常過程の場合には $X(n) - m_X$ を考えることで平均を 0 にできるので、この仮定によって一般性を失いません。

2.4 相関行列の性質

相関行列は多くの信号処理手法において中心的な役割を果たすため、以下では式 (2.44) で定義される相関行列のいくつかの性質を説明します。その性質の多くが、相関行列がエルミート行列であることに由来している点に注意しましょう。

性質 1：相関行列はエルミート行列

相関行列の定義の式 (2.44) より

$$\begin{aligned} \mathbf{R}_{\mathbf{X}}^H &= (E[\mathbf{X}\mathbf{X}^H])^H \\ &= E[(\mathbf{X}\mathbf{X}^H)^H] \end{aligned}$$

$$= E[\mathbf{X}\mathbf{X}^H]$$

$$= \mathbf{R}_{\mathbf{X}}$$

となることから簡単に確認できます。

なお、広義定常過程の自己相関行列の場合には、式(2.56)の成分が

$$\begin{aligned} r_X(-k) &= E[X(n)X^*(n+k)] \\ &= (E[X^*(n)X(n+k)])^* \\ &= (E[X^*(n-k)X(n)])^* \\ &= r_X^*(k) \end{aligned}$$

となることを用いて

$$E[\mathbf{X}(n)\mathbf{X}^H(n)] = \begin{bmatrix} r_X(0) & r_X(1) & \dots & r_X(N-1) \\ r_X^*(1) & r_X(0) & \dots & r_X(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_X^*(N-1) & r_X^*(N-2) & \dots & r_X(0) \end{bmatrix}$$

と書くことでも、性質1が確認できます。

性質2：相関行列は非負定値

エルミート行列 $\mathbf{A} \in \mathbb{C}^{N \times N}$ が**非負定値** (non-negative definite) であるとは、任意の非零ベクトル $\mathbf{u} \in \mathbb{C}^N$ に対してエルミート形式 (Hermitian form, 実数の場合は**2次形式** (quadratic form)) $\mathbf{u}^H \mathbf{A} \mathbf{u}$ が非負になることをいいます。非負定値は**半正定値** (positive semidefinite) ともいいます。また、任意の \mathbf{u} に対してエルミート形式が正となるエルミート行列は**正定値** (positive definite), 負となる行列は**負定値** (negative definite) といいます。

値の正負について考えることは実数に対して有効なものなので、成分が複素数の行列 \mathbf{A} とベクトル \mathbf{u} から計算される $\mathbf{u}^H \mathbf{A} \mathbf{u}$ について、その値の正負を議

論することに疑問をもたれるかもしれません。実は、エルミート行列 \mathbf{A} のエルミート形式は

$$(\mathbf{u}^H \mathbf{A} \mathbf{u})^* = (\mathbf{u}^H \mathbf{A} \mathbf{u})^H = \mathbf{u}^H \mathbf{A}^H \mathbf{u} = \mathbf{u}^H \mathbf{A} \mathbf{u}$$

となり、その値がつねに実数になることが示されるので、その正負について考えることができるのです。

$\mathbf{u} \in \mathbb{C}^N$ を任意の確定的な非零ベクトルとし、 \mathbf{X} との内積を $Y = \mathbf{u}^H \mathbf{X}$ とします。このとき

$$\begin{aligned} E[|Y|^2] &= E[YY^*] \\ &= E[\mathbf{u}^H \mathbf{X} \mathbf{X}^H \mathbf{u}] \\ &= \mathbf{u}^H E[\mathbf{X} \mathbf{X}^H] \mathbf{u} \\ &= \mathbf{u}^H \mathbf{R}_X \mathbf{u} \end{aligned}$$

となります。 $|Y|^2$ は \mathbf{X} の任意の実現値に対して実数で $|y|^2 \geq 0$ であることから、その集合平均も実数で $E[|Y|^2] \geq 0$ となります。したがって、上記の性質 2 が示されました。

性質 3：相関行列の固有値はすべて非負の実数

相関行列 \mathbf{R}_X の固有値 (eigenvalue) を $\lambda_1, \lambda_2, \dots, \lambda_N$ とし、固有値 λ_i に対応する固有ベクトル (eigenvector) を $\mathbf{q}_i = [q_{i,1} \ q_{i,2} \ \cdots \ q_{i,N}]^T (\neq \mathbf{0})$ とすると、固有値の定義より

$$\mathbf{R}_X \mathbf{q}_i = \lambda_i \mathbf{q}_i \quad (2.58)$$

が成り立ちます。両辺に左から \mathbf{q}_i^H を乗算すると

$$\mathbf{q}_i^H \mathbf{R}_X \mathbf{q}_i = \lambda_i \mathbf{q}_i^H \mathbf{q}_i$$

となります。さらに $\mathbf{q}_i^H \mathbf{q}_i$ は

$$\mathbf{q}_i^H \mathbf{q}_i = \sum_{j=1}^N |q_{i,j}|^2$$

より正の実数であることがわかるので、これで両辺を割って

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R}_X \mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i} \quad (2.59)$$

を得ます。

性質 2 より分子の $\mathbf{q}_i^H \mathbf{R}_X \mathbf{q}_i$ は非負の実数値をとるので、 λ_i は非負の実数となります。なお、エルミート行列のエルミート形式が実数であることから、一般のエルミート行列の固有値は実数になる（ただし、非負とは限らない）ことにも注意します。

式 (2.59) の右辺の比は行列 \mathbf{R}_X とベクトル \mathbf{q}_i に対するレイリー商 (Rayleigh quotient) と呼ばれます。したがって、行列 \mathbf{R}_X の固有値 λ_i は、行列 \mathbf{R}_X と対応する固有ベクトル \mathbf{q}_i に対するレイリー商ということができます。

性質 4：相関行列の相異なる固有値に対応する固有ベクトルは直交する

相関行列 \mathbf{R}_X の固有値 $\lambda_1, \lambda_2, \dots, \lambda_N$ がすべて異なるとし、固有値 λ_i に対応する固有ベクトルを \mathbf{q}_i とします。式 (2.58) の両辺に左から $\mathbf{q}_j^H, j \neq i$ を乗算することで

$$\mathbf{q}_j^H \mathbf{R}_X \mathbf{q}_i = \lambda_i \mathbf{q}_j^H \mathbf{q}_i \quad (2.60)$$

を得ます。一方

$$\mathbf{R}_X \mathbf{q}_j = \lambda_j \mathbf{q}_j \quad (2.61)$$

の両辺のエルミート転置をとると、性質 1 より \mathbf{R}_X がエルミート行列であること、また性質 3 より λ_j が実数であることから

$$\mathbf{q}_j^H \mathbf{R}_X = \lambda_j \mathbf{q}_j^H \quad (2.62)$$

となります。この両辺に右から \mathbf{q}_i を乗算して

$$\mathbf{q}_j^H \mathbf{R}_X \mathbf{q}_i = \lambda_j \mathbf{q}_j^H \mathbf{q}_i \quad (2.63)$$

を得ます。さらに、式 (2.60) と式 (2.63) を辺々引き算することで

$$(\lambda_i - \lambda_j) \mathbf{q}_j^H \mathbf{q}_i = 0 \quad (2.64)$$

を得ます。仮定より $i \neq j$ に対して $\lambda_i \neq \lambda_j$ なので

$$\mathbf{q}_j^H \mathbf{q}_i = 0, \quad i \neq j \quad (2.65)$$

が示されました。

なお、性質 4 を示す際に、相関行列 \mathbf{R}_X がエルミート行列であることとその固有値が実数であることしか用いていないことに注意します。したがって、性質 4 は相関行列だけでなく、一般のエルミート行列について成立する性質です。

性質 5：相関行列はユニタリ行列を用いて対角化される

相関行列 \mathbf{R}_X の固有値 $\lambda_1, \lambda_2, \dots, \lambda_N$ に対応する固有ベクトルをそれぞれ $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N$ とし、固有ベクトルは正規化されているもの、すなわち、 $\|\mathbf{q}_i\|_2 \triangleq \sqrt{\mathbf{q}_i^H \mathbf{q}_i} = 1$ とします。

まず、固有値がすべて異なる場合について考えます。このとき、すべての i についての式 (2.58) を並べることで

$$\mathbf{R}_X \mathbf{Q} = \mathbf{Q} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} \quad (2.66)$$

を得ます。ここで、 \mathbf{Q} は

$$\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_N] \quad (2.67)$$

で定義され、性質 4 より

$$Q^H Q = I_N \quad (2.68)$$

が成り立つことからユニタリ行列 (unitary matrix) です。式 (2.66) の両辺に左から Q^H を乗算することで

$$Q^H R_X Q = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} \quad (2.69)$$

が得られます。

後述する部分空間法で考える状況のように、固有値に同じ値をもつものがある場合（固有値が縮退している (degenerate) といいます）は、重複する固有値の数と同じ次元で、そこに含まれる任意のベクトルがその固有値をもつ固有ベクトルになるような部分空間 (subspace)[†]が存在します。この部分空間内の固有ベクトルと他の固有値に対応する固有ベクトルは性質 4 から直交します。また、同じ固有値をもつ固有ベクトルの部分空間にはいつでも直交基底 (orthogonal basis) をとることができます。このため、固有値が重複している場合には、その固有値に対応する部分空間の直交基底を q_i としてとることでいつでも Q がユニタリ行列になるようにでき、性質 5 が示されます。ただし、その表現は一意的ではないことに注意します。より詳細な説明は、文献11)などを参照してください（実対称行列について説明されていますが、エルミート行列の場合にも同様の議論が成り立ちます）。なお、性質 5 も相関行列だけでなく、一般のエルミート行列について成立する性質です。

[†] ベクトル空間 V の部分集合を W としたとき、任意の $a, b \in W$ に対して $a + b \in W$ であり、かつ任意のスカラー c と $a \in W$ に対して $ca \in W$ となるとき、 W を部分空間といいます。

性質 6：相異なる固有値をもつ相関行列 \mathbf{R}_X の最大固有値 λ_{\max} は、最適化問題

$$\lambda_{\max} = \max_{\|q\|_2=1} q^H \mathbf{R}_X q \quad (2.70)$$

によって得られる。

相関行列 \mathbf{R}_X の固有値を降順に $\lambda_1 > \lambda_2 > \dots > \lambda_N$ とし、対応する正規化固有ベクトルをそれぞれ $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N$ とします。最適化問題の最大化ベクトルを \mathbf{p} とすると、性質 4 より $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N$ は N 次元ベクトル空間の正規直交基底 (orthonormal basis) であることから、 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N$ の線形結合により

$$\mathbf{p} = \sum_{i=1}^N \alpha_i \mathbf{q}_i \quad (2.71)$$

と書けます。ただし、 $\alpha_i, i = 1, 2, \dots, N$ は複素数のスカラーです。

性質 5 より

$$\begin{aligned} \mathbf{R}_X &= Q \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} Q^H \\ &= \sum_{i=1}^N \lambda_i \mathbf{q}_i \mathbf{q}_i^H \end{aligned} \quad (2.72)$$

となることから

$$\begin{aligned} \mathbf{p}^H \mathbf{R}_X \mathbf{p} &= \mathbf{p}^H \left(\sum_{i=1}^N \lambda_i \mathbf{q}_i \mathbf{q}_i^H \right) \mathbf{p} \\ &= \sum_{i=1}^N \lambda_i \mathbf{p}^H \mathbf{q}_i \mathbf{q}_i^H \mathbf{p} \end{aligned} \quad (2.73)$$

となります。さらに、性質 4 を用いると

$$\mathbf{p}^H \mathbf{q}_i = \alpha_i^* \quad (2.74)$$

$$\mathbf{q}_i^H \mathbf{p} = \alpha_i \quad (2.75)$$

となるので

$$\mathbf{p}^H \mathbf{R}_X \mathbf{p} = \sum_{i=1}^N \lambda_i |\alpha_i|^2 \quad (2.76)$$

と書けます。ここで、仮定より $\lambda_1 > \lambda_2 > \dots > \lambda_N$ なので、 $\mathbf{p}^H \mathbf{R}_X \mathbf{p}$ は

$$\mathbf{p}^H \mathbf{R}_X \mathbf{p} \leq \lambda_1 \sum_{i=1}^N |\alpha_i|^2 \quad (2.77)$$

と上から押さえられます。等号成立は $i \neq 1$ なるすべての i について $|\alpha_i| = 0$ のときです。また、 \mathbf{p} は正規化されているのでこのとき $|\alpha_1| = 1$ となり、最終的に

$$\max_{\|\mathbf{q}\|_2=1} \mathbf{q}^H \mathbf{R}_X \mathbf{q} = \lambda_1 \quad (2.78)$$

を得ます。

なお、ここでは簡単のため \mathbf{R}_X が相異なる固有値をもつ場合について考えましたが、固有値が重複している場合も同様の手順で最大固有値が求まります。さらに、同様の手順を繰り返すことで最大固有値だけでなく、他の固有値も求めることができます。詳細は文献12), 13)などを参照してください。なお、性質6も相関行列だけでなく、一般のエルミート行列について成立する性質です。

2.5 条件付き独立

二つの確率変数 X, Y が独立であることの定義はすでに述べましたが、統計的信号処理では、多くの場合、何らかの観測に基づいて未知の量を推定するので、 X, Y とは別のある確率変数 Z が観測されたという条件の下で X, Y が独立であるかどうかを考えることが重要です。特に、確率推論の問題では、後述

する周辺事後確率 (marginal posterior probability) の計算の際に、この**条件付き独立性** (conditional independence) を利用することが肝となります。

離散確率変数 X, Y, Z の同時確率分布を $P(x, y, z)$ とすると（連続確率変数の場合には、同時分布の確率密度関数 $p(x, y, z)$ とする）[†]、乗法定理と周辺化により

$$P(x, y|z) = \frac{P(x, y, z)}{P(z)} \quad (2.79)$$

$$P(x|z) = \frac{P(x, z)}{P(z)} \quad (2.80)$$

$$P(y|z) = \frac{P(y, z)}{P(z)} \quad (2.81)$$

となります。このとき、すべての x, y, z の値について

$$P(x, y|z) = P(x|z)P(y|z) \quad (2.82)$$

が成り立つとき、確率変数 X と Y は Z を与えた下で条件付き独立といい、 $X \perp\!\!\!\perp Y|Z$ と表記します。例えば、 $X \perp\!\!\!\perp Y|Z$ が成り立つとき

$$P(x, y, z) = \frac{P(x, z)P(y, z)}{P(z)} \quad (2.83)$$

$$P(x|y, z) = P(x|z) \quad (2.84)$$

などが示されます。

複数の確率変数の同時確率分布が与えられたときに、ある確率変数が与えられた下でどの確率変数とどの確率変数が条件付き独立であるかを式の上から判定することはそれほど簡単な問題ではありません。しかしながら、つぎに述べるグラフィカルモデルを用いると、条件付き独立性をグラフの構造からただちに読みとることができます。

[†] 本節以降、表記が煩雑になるのを避けるために $P_{XYZ}(x, y, z)$ や $p_{XYZ}(x, y, z)$ などの下付きの文字を省略して、単に $P(x, y, z)$ や $p(x, y, z)$ と表記することにします。

2.6 グラフィカルモデル

グラフィカルモデル (graphical model) とは多変量の確率変数の関係をグラフによって表現するもので、さまざまな表現法が提案されています。ここでは、確率推論の問題を扱う際に利用する代表的なグラフィカルモデルとして、ベイジアンネットワークとファクターグラフについて説明します。

2.6.1 ベイジアンネットワーク

ベイジアンネットワーク (Bayesian network) はつぎのように定義されます（グラフを定義するには、ノードとエッジの定義を与えればよいことに注意します）。

定義 2.1 (ベイジアンネットワーク) 確率変数 X_1, X_2, \dots, X_n の同時確率分布 $P(x_1, x_2, \dots, x_n)$ に対応するベイジアンネットワークはつぎの性質を満たす**有向非巡回グラフ** (directed acyclic graph, DAG), すなわち矢印の向きにたどって同じノードに戻ることがない有向グラフである。

1. 確率変数が各ノードに対応
2. 同時確率分布が

$$P(x_1, x_2, \dots, x_n) = P(x_1|S_1)P(x_2|S_2) \cdots P(x_n|S_n)$$

という形で与えられるとき（これを同時確率分布の**因数分解**あるいは**因子分解** (factorization) と呼ぶ）、 S_i が x_i の親ノードの集合（つまり、 $x_j \in S_i \rightarrow x_i$ に対応する有向エッジが存在）

ベイジアンネットワークの具体例として、同時確率分布が

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2|x_1)P(x_3|x_4)P(x_4|x_1)P(x_5|x_4) \quad (2.85)$$

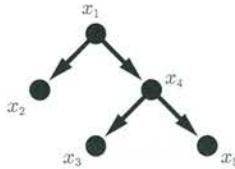


図 2.1 ベイジアンネットワークの例

で与えられる場合を考えると、対応するベイジアンネットワークは図 2.1 のようになります。

また、確率変数 X_1, X_2, \dots, X_n のマルコフ連鎖 (Markov chain) の同時確率分布は

$$P(x_1, x_2, \dots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_2)\cdots P(x_n|x_{n-1}) \quad (2.86)$$

で与えられるため、そのベイジアンネットワークは図 2.2 のようになります。



図 2.2 ベイジアンネットワークの例 (マルコフ連鎖)

逆に、図 2.1 や図 2.2 のベイジアンネットワークが与えられると、式 (2.85) や式 (2.86) の同時確率分布が再現できることに注意します。

ベイジアンネットワークと条件付き独立の関係を理解するために、図 2.3 に示される確率変数 X, Y, Z についての三つのベイジアンネットワークについて考えてみましょう。

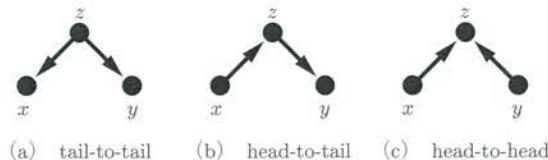


図 2.3 ベイジアンネットワークと条件付き独立

図 2.3(a) に示されるペイジアンネットワークは、中間にあるノード z に両方の矢印の後ろ側 (tail) が向いているので tail-to-tail と呼ばれ、これに対応する同時確率分布は

$$P(x, y, z) = P(x|z)P(y|z)P(z) \quad (2.87)$$

です。これより

$$P(x, y) = \sum_z P(x|z)P(y|z)P(z) \quad (2.88)$$

は一般に $P(x)P(y)$ とならないため、 X と Y は独立ではありません。一方

$$P(x, y|z) = \frac{P(x, y, z)}{P(z)} \quad (2.89)$$

$$= \frac{P(x|z)P(y|z)P(z)}{P(z)} \quad (2.90)$$

$$= P(x|z)P(y|z) \quad (2.91)$$

となるため、 X と Y は Z を与えた下で条件付き独立です。

図 2.3(b) に示されるペイジアンネットワークは、中間にあるノード z に一方の矢印の頭 (head) ともう一方の後ろ側が向いているので head-to-tail と呼ばれ、これに対応する同時確率分布は

$$P(x, y, z) = P(x)P(y|z)P(z|x) \quad (2.92)$$

です。これより

$$P(x, y) = \sum_z P(x)P(y|z)P(z|x) \quad (2.93)$$

$$= P(x) \sum_z P(y|z)P(z|x) \quad (2.94)$$

は一般に $P(x)P(y)$ とならないため、 X と Y は独立ではありませんが

$$P(x, y|z) = \frac{P(x)P(y|z)P(z|x)}{P(z)} \quad (2.95)$$

$$= \frac{P(x, z)P(y|z)}{P(z)} \quad (2.96)$$

$$= P(x|z)P(y|z) \quad (2.97)$$

となるため、 X と Y は Z を与えた下で条件付き独立です。

最後に、図 2.3(c) に示されるベイジアンネットワークは、中間にあるノード z に両方の矢印の頭が向いているので head-to-head と呼ばれ、これに対応する同時確率分布は

$$P(x, y, z) = P(x)P(y)P(z|x, y) \quad (2.98)$$

です。これより

$$P(x, y) = \sum_z P(x)P(y)P(z|x, y) \quad (2.99)$$

$$= P(x)P(y) \quad (2.100)$$

となり、 X と Y は独立ですが

$$P(x, y|z) = \frac{P(x)P(y)P(z|x, y)}{P(z)} \quad (2.101)$$

は一般に $P(x|z)P(y|z)$ とならず、 X と Y は Z を与えた下で条件付き独立ではありません。

以上のように、ベイジアンネットワークの構造 (tail-to-tail, head-to-tail) から条件付き独立性を読みとることが可能です。例えば、図 2.1 の例では、ノード x_1, x_2, x_4 や x_3, x_4, x_5 の間に tail-to-tail の構造があるので

$$X_2 \perp\!\!\!\perp X_4 | X_1 \quad (2.102)$$

$$X_3 \perp\!\!\!\perp X_5 | X_4 \quad (2.103)$$

がただちにわかります。また、図 2.2 のマルコフ連鎖の例では、ノード x_i, x_{i+1}, x_{i+2} の間に head-to-tail の構造があるので

$$X_i \perp\!\!\!\perp X_{i+2} | X_{i+1}, \quad i = 1, 2, \dots, n-2 \quad (2.104)$$

がただちに読みとれます。

確率変数は複数をまとめて一つの確率変数として考えることができるので、ベイジアンネットワークの中に tail-to-tail といった構造を見つける際に、複数のノードをまとめて一つと考えることができます。例えば、図 2.1 の例ではノード x_1, x_4 をまとめて一つと考えると x_2, x_5 と x_1, x_4 は tail-to-tail の構造になっているので

$$X_2 \perp\!\!\!\perp X_5 | X_1, X_4 \quad (2.105)$$

がわかります。実際

$$\begin{aligned} P(x_2, x_5 | x_1, x_4) &= \frac{P(x_1, x_2, x_4, x_5)}{P(x_1, x_4)} \\ &= \frac{\sum_{x_3} P(x_1, x_2, x_3, x_4, x_5)}{P(x_1, x_4)} \\ &= \frac{\sum_{x_3} P(x_1)P(x_2|x_1)P(x_3|x_4)P(x_4|x_1)P(x_5|x_4)}{P(x_1, x_4)} \\ &= \frac{P(x_1)P(x_2|x_1)P(x_4|x_1)P(x_5|x_4)\sum_{x_3} P(x_3|x_4)}{P(x_1, x_4)} \end{aligned}$$

となり、ここで $P(x_1, x_4) = P(x_1)P(x_4|x_1)$ と $\sum_{x_3} P(x_3|x_4) = 1$ を用いると

$$P(x_2, x_5 | x_1, x_4) = P(x_2|x_1)P(x_5|x_4) \quad (2.106)$$

を得ます。さらに、ベイジアンネットワークの構造からわかるように、条件付き独立性

$$P(x_2, x_4 | x_1) = P(x_2|x_1)P(x_4|x_1) \quad (2.107)$$

$$P(x_1, x_5 | x_4) = P(x_1|x_4)P(x_5|x_4) \quad (2.108)$$

が成立することを用いると

$$\begin{aligned} P(x_2, x_5 | x_1, x_4) &= \frac{P(x_2, x_4 | x_1)}{P(x_4 | x_1)} \frac{P(x_1, x_5 | x_4)}{P(x_1 | x_4)} \\ &= P(x_2 | x_1, x_4) P(x_5 | x_1, x_4) \end{aligned} \quad (2.109)$$

が得られます。

2.6.2 ファクターグラフ

後述する確率伝播法のsum-productアルゴリズムは、**ファクターグラフ**(factor graph)と呼ばれるグラフ上の**メッセージ伝播**(message passing)アルゴリズムとして記述されます。ファクターグラフは確率分布に限らず、より一般の**多変数関数**(multivariable function)の因数分解を表す無向グラフで、2種類のノード、すなわち関数ノードと変数ノードから構成されます。

変数の集合を $\{x_1, x_2, \dots, x_n\}$ とし、多変数関数が

$$f(x_1, x_2, \dots, x_n) = f_1(A_1)f_2(A_2) \cdots f_m(A_m) \quad (2.110)$$

のように因数分解されるとします。ただし、 A_1, A_2, \dots, A_m はすべての変数の集合 $\{x_1, x_2, \dots, x_n\}$ の部分集合です。このとき、関数ノードと変数ノードは、それぞれローカル関数 $f_i(A_i)$ と変数 x_j に対応します。そして、 $f_i(A_i)$ に対応する関数ノードと A_i に含まれる変数に対応する変数ノードとが無向エッジで接続されることによりグラフが構成されます(図 2.4)。



図 2.4 ファクターグラフの接続ルール

ファクターグラフの具体例として、同時確率分布が式(2.85)で与えられる場合について考えてみます。式(2.85)の同時確率分布を変数が x_1, x_2, \dots, x_5 の多変数関数 $f(x_1, x_2, x_3, x_4, x_5)$ とみなし、ローカル関数を

$$\begin{aligned}
 f_1(x_1) &= P(x_1) \\
 f_2(x_1, x_2) &= P(x_2|x_1) \\
 f_3(x_1, x_4) &= P(x_4|x_1) \\
 f_4(x_3, x_4) &= P(x_3|x_4) \\
 f_5(x_4, x_5) &= P(x_5|x_4)
 \end{aligned}$$

と定義して

$$f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1)f_2(x_1, x_2)f_4(x_3, x_4)f_3(x_1, x_4)f_5(x_4, x_5) \quad (2.111)$$

とすることで、ファクターグラフは図 2.5 のようになります。

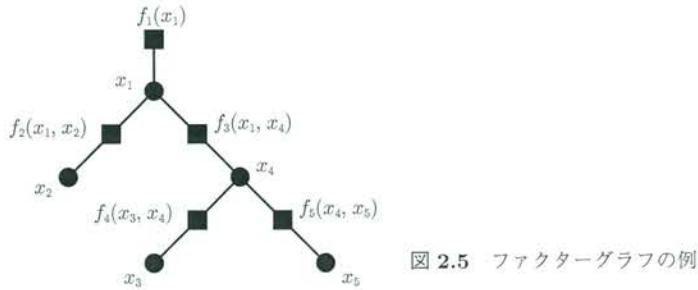


図 2.5 ファクターグラフの例

同時確率分布が式 (2.86) で与えられるマルコフ連鎖のファクターグラフを描くために、同時確率分布を変数が x_1, x_2, \dots, x_n の多変数関数とみなして

$$\begin{aligned}
 f_1(x_1) &= P(x_1) \\
 f_i(x_{i-1}, x_i) &= P(x_i|x_{i-1}), \quad i = 2, 3, \dots, n
 \end{aligned}$$

と対応づけると、同時確率分布は

$$f(x_1, x_2, \dots, x_n) = f_1(x_1)f_2(x_1, x_2)f_3(x_2, x_3)\cdots f_n(x_{n-1}, x_n) \quad (2.112)$$

と書けます。これよりマルコフ連鎖のファクターグラフは図 2.6 のようになります。

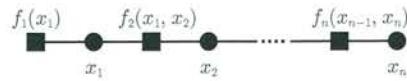


図 2.6 ファクターグラフの例
(マルコフ連鎖)

2.7 むすび

本章では、信号処理の基礎を理解する上で必要最小限の確率および統計に関する基礎事項について説明しました。さらに確率・統計について学びたい方は文献14)をお勧めします。確率・統計の教科書は初等的な確率論に基づく内容がやや平易すぎるものと、測度論(measure theory)に基づく非常に難解なものに二極化している印象がありますが、本書は情報系分野の研究者に必要とされる内容がバランスよくコンパクトにまとめられていながら、工学部の学生が読み進めていけるように配慮されている点が素晴らしいです。統計的信号処理のほとんどの場面で、この文献の内容を理解すれば十分であると思われますが、やはりきちんと確率論を学びたいという方は文献15)がより高度な確率論への橋渡しにお勧めです。特に、 σ -集合体(σ -field)を導入する必要性について詳しく書かれた第9章は、工学分野の方にとって大変有益であると思われます。

3

ウィルティンガー微分

Next SIP

工学部の標準的なカリキュラムで学ぶ複素関数論では、複素微分可能な関数（正則関数）についてのみ議論されることがほとんどですが、通信の信号処理では実微分可能であるが複素微分可能ではない関数の最小化などについて考える必要があります。実際、通信システムの信号処理で出てくるコスト関数は、多くの場合、複素数の引数をもつが実数値をとる関数となります。これは実微分可能であるが複素微分可能ではない関数の典型例です。本章では、そのような関数の複素数、複素ベクトル、複素行列による微分を形式的に定義可能な手法である **ウィルティンガー微分** (Wirtinger derivative) について説明します。通信の信号処理の勉強をするつもりが解析学の話が出てきて戸惑われるかもしれません。ウィルティンガー微分を利用するとコスト関数の勾配が非常に簡単に求まるのでぜひマスターしましょう。

3.1 実関数の微分

3.1.1 実1変数関数の微分

まず、一つの実数の引数をもつ関数の微分について復習しましょう。

定義 3.1 (実1変数関数の微分)¹⁶⁾ $D \subseteq \mathbb{R}$ を関数 $f: D \rightarrow \mathbb{R}$ の定義域 (domain) とする。

$$\lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (3.1)$$

が任意の内点 $x \in D$ について存在するとき、関数 f は**微分可能** (differentiable) という。

ここで、「極限が存在する」とは Δx を 0 に近づける際の近づけ方によらず極限の値が一つに定まることをいいます。

3.1.2 実関数の偏微分

つぎに、実 2 変数関数 $f(x, y)$ の微分について考えましょう。2 変数関数では変化させられる量が二つあるので、片方の変数だけ動かす場合と両方の変数を動かす場合を考えられます。片方だけを動かしたときの変化量を考える微分は**偏微分** (partial derivative) と呼ばれます。

定義 3.2 (実 2 変数関数の偏微分)¹⁷⁾ $D \subseteq \mathbb{R}^2$ を関数 $f : D \rightarrow \mathbb{R}$ の定義域とする。

$$\frac{\partial f(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \quad (3.2)$$

$$\frac{\partial f(x, y)}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \quad (3.3)$$

が任意の内点 $(x, y) \in D$ について存在するとき、これらを偏微分という。

偏微分が存在するならば以下が成り立ちます。

$$f(x + \Delta x, y) = f(x, y) + \frac{\partial f(x, y)}{\partial x} \Delta x + \epsilon_x \Delta x \quad (3.4)$$

$$f(x, y + \Delta y) = f(x, y) + \frac{\partial f(x, y)}{\partial y} \Delta y + \epsilon_y \Delta y \quad (3.5)$$

ただし、 $\Delta x \rightarrow 0$ のとき $\epsilon_x \rightarrow 0$ で、 $\Delta y \rightarrow 0$ のとき $\epsilon_y \rightarrow 0$ です。

各成分が偏微分で与えられるベクトルを**勾配ベクトル** (gradient vector) と呼びます。偏微分の定義は 3 変数以上の場合にも容易に拡張できるので、例えば、実 N 変数のベクトル $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]^T \in \mathbb{R}^N$ を引数にもつ関数 $J(\mathbf{w})$

の勾配は

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial J(\mathbf{w})}{\partial w_1} & \frac{\partial J(\mathbf{w})}{\partial w_2} & \dots & \frac{\partial J(\mathbf{w})}{\partial w_N} \end{bmatrix} \quad (3.6)$$

で与えられます。

勾配ベクトルは信号処理できわめて重要な役割を果たします。例えば、 \mathbf{w} がシステムを特徴づけるパラメータ (parameter) で、 $J(\mathbf{w})$ がコスト関数 (cost function)，すなわちその値を小さくしたい関数である場合には

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0} \quad (3.7)$$

を満足する停留点 (stationary point) \mathbf{w} を求ることで、 $J(\mathbf{w})$ の極小値 (local minimum) を与える点の候補を求めることができます。 $J(\mathbf{w})$ が凸関数 (convex function)[†]の場合には、停留点 \mathbf{w} は $J(\mathbf{w})$ の大域的な最小値 (global minimum) を与えます。これは多くの信号処理技術で利用される勾配法 (gradient method) の基本的な考え方です。

なお、勾配ベクトルを列ベクトル (column vector) と行ベクトル (row vector) のどちらで定義するかは文献によって異なります。本書では、後に複素関数の勾配を考える際に表記がシンプルになるという理由で、勾配ベクトルを行ベクトルで定義することにします。

3.1.3 実関数の全微分

つぎに、実2変数関数 $f(x, y)$ に戻って、 x, y をそれぞれ $\Delta x, \Delta y$ だけ変化させたときの $f(x, y)$ からの変動分 Δf 、すなわち

$$f(x + \Delta x, y + \Delta y) = f(x, y) + \Delta f \quad (3.8)$$

を考えます。 x についての偏微分の式 (3.4) において、 y に $y + \Delta y$ を代入することで

[†] 関数 $f(\mathbf{x})$ が凸関数であるとは、その定義域 \mathcal{X} が凸集合 (convex set) であり、任意の $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ と $0 \leq \lambda \leq 1$ について、 $f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2)$ であることをいいます。また、 \mathcal{X} が凸集合であるとは、任意の $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ と $0 \leq \lambda \leq 1$ について、 $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in \mathcal{X}$ であることをいいます。

$$\begin{aligned}
 & f(x + \Delta x, y + \Delta y) \\
 &= f(x, y + \Delta y) + \frac{\partial f(x, y + \Delta y)}{\partial x} \Delta x + \epsilon_x \Delta x \\
 &= f(x, y) + \frac{\partial f(x, y)}{\partial y} \Delta y + \left\{ \frac{\partial f(x, y)}{\partial x} + \frac{\partial}{\partial y} \left(\frac{\partial f(x, y)}{\partial x} \right) \Delta y + \epsilon_{dy} \Delta y \right\} \\
 &\quad \cdot \Delta x + \epsilon_x \Delta x + \epsilon_y \Delta y
 \end{aligned} \tag{3.9}$$

となります。ここで、 $\Delta x \rightarrow 0$ 、 $\Delta y \rightarrow 0$ とし、 $\Delta x \Delta y$ の項を無視して、さらに Δ を d で置き換えると

$$df = \frac{\partial f(x, y)}{\partial x} dx + \frac{\partial f(x, y)}{\partial y} dy \tag{3.10}$$

を得ます。これを**全微分** (total derivative) と呼びます。

全微分の式 (3.10) の導出では暗黙のうちにいくつかの仮定をしていましたが、全微分可能条件はつぎで与えられます¹⁷⁾。

定理 3.1 (全微分可能条件) 関数 $f(x, y)$ は領域 $D \subseteq \mathbb{R}^2$ で連続で、偏微分 $\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y}$ が至るところで定義されていて、さらに $\frac{\partial f(x, y)}{\partial x}$ または $\frac{\partial f(x, y)}{\partial y}$ は点 $(x, y) \in D$ で連続であるとする。このとき

$$\begin{aligned}
 \Delta f &= f(x + \Delta x, y + \Delta y) - f(x, y) \\
 &= \frac{\partial f(x, y)}{\partial x} \Delta x + \frac{\partial f(x, y)}{\partial y} \Delta y + \epsilon_x \Delta x + \epsilon_y \Delta y
 \end{aligned} \tag{3.11}$$

が成り立つ。ただし、 ϵ_x, ϵ_y は $\Delta x, \Delta y \rightarrow 0$ のとき $\epsilon_x, \epsilon_y \rightarrow 0$ となる量である。

定理 3.1 はつぎのように確認できます。まず、仮定より $f(x, y)$ の偏微分が存在することから

$$\Delta f = f(x + \Delta x, y + \Delta y) - f(x, y)$$

$$\begin{aligned}
 &= f(x + \Delta x, y + \Delta y) - f(x, y + \Delta y) + f(x, y + \Delta y) - f(x, y) \\
 &= \frac{\partial f(x, y + \Delta y)}{\partial x} \Delta x + \epsilon_x \Delta x + \frac{\partial f(x, y)}{\partial y} \Delta y + \epsilon_y \Delta y
 \end{aligned} \tag{3.12}$$

となります。ここで、 $\frac{\partial f(x, y)}{\partial x}$ が (x, y) で連続なら、 $\Delta y \rightarrow 0$ のとき

$$\epsilon = \frac{\partial f(x, y + \Delta y)}{\partial x} - \frac{\partial f(x, y)}{\partial x} \rightarrow 0 \tag{3.13}$$

となります。これより

$$\begin{aligned}
 \Delta f &= \left\{ \frac{\partial f(x, y)}{\partial x} + \epsilon \right\} \Delta x + \epsilon_x \Delta x + \frac{\partial f(x, y)}{\partial y} \Delta y + \epsilon_y \Delta y \\
 &= \frac{\partial f(x, y)}{\partial x} \Delta x + (\epsilon + \epsilon_x) \Delta x + \frac{\partial f(x, y)}{\partial y} \Delta y + \epsilon_y \Delta y
 \end{aligned} \tag{3.14}$$

となり、 $\Delta x, \Delta y \rightarrow 0$ のとき全微分の式が得られることがわかります。なお、ここでは $\frac{\partial f(x, y)}{\partial x}$ が (x, y) で連続であると仮定しましたが、 $\frac{\partial f(x, y)}{\partial y}$ が連続の場合も同様に確認できます。

全微分の定義も 3 変数以上の場合に容易に拡張できます。 N 変数の実ベクトル $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]^T \in \mathbb{R}^N$ を引数にもつ関数 $J(\mathbf{w})$ の全微分は、勾配ベクトルを用いて

$$\begin{aligned}
 dJ &= \frac{\partial J(\mathbf{w})}{\partial w_1} dw_1 + \frac{\partial J(\mathbf{w})}{\partial w_2} dw_2 + \cdots + \frac{\partial J(\mathbf{w})}{\partial w_N} dw_N \\
 &= \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} d\mathbf{w} \\
 &= (d\mathbf{w})^T \left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right)^T
 \end{aligned} \tag{3.15}$$

と書けます。ただし、 $d\mathbf{w} = [dw_1 \ dw_2 \ \cdots \ dw_N]^T \in \mathbb{R}^N$ です。

3.1.4 全微分による勾配の計算

勾配ベクトルが信号処理で重要な役割を果たすことはすでに述べましたが、関数によっては勾配を直接求めることが難しい場合があります。そのような場

合でも、全微分 dJ を求めることができると、式(3.15)の関係、あるいはこれの両辺のトレースをとった

$$dJ = \text{tr} \left[\left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right)^T (d\mathbf{w})^T \right] \quad (3.16)$$

なる関係を利用することで、形式的に勾配 $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$ を求めることができます。

例えば、 $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]^T \in \mathbb{R}^N$ として

$$J(\mathbf{w}) = \|\mathbf{w}\|_2^2 = \mathbf{w}^T \mathbf{w} = \text{tr}[\mathbf{w} \mathbf{w}^T] \quad (3.17)$$

の勾配を全微分から求めてみましょう。

まず、全微分とトレースの順序が交換可能である性質を用いると

$$dJ = d\text{tr}[\mathbf{w} \mathbf{w}^T] = \text{tr}[d(\mathbf{w} \mathbf{w}^T)] \quad (3.18)$$

となります。ここで

$$(\mathbf{w} + \Delta \mathbf{w})(\mathbf{w} + \Delta \mathbf{w})^T - \mathbf{w} \mathbf{w}^T = (\Delta \mathbf{w})\mathbf{w}^T + \mathbf{w}(\Delta \mathbf{w})^T + \Delta \mathbf{w}(\Delta \mathbf{w})^T \quad (3.19)$$

より行列 $\mathbf{w} \mathbf{w}^T$ の全微分[†]は

$$d(\mathbf{w} \mathbf{w}^T) = (d\mathbf{w})\mathbf{w}^T + \mathbf{w}(d\mathbf{w})^T \quad (3.20)$$

となります。したがって

$$dJ = \text{tr}[(d\mathbf{w})\mathbf{w}^T + \mathbf{w}(d\mathbf{w})^T] \quad (3.21)$$

$$= \text{tr}[\mathbf{w}(d\mathbf{w})^T + \mathbf{w}(d\mathbf{w})^T] \quad (3.22)$$

$$= \text{tr}[2\mathbf{w}(d\mathbf{w})^T] \quad (3.23)$$

[†] $F(\mathbf{w}) = \mathbf{w} \mathbf{w}^T$ はベクトル引数行列値関数ですが、スカラー値関数の全微分の式(3.10)を求めたときと同様に、 $\Delta F = F(\mathbf{w} + \Delta \mathbf{w}) - F(\mathbf{w})$ を考えて $\Delta \mathbf{w}$ の 2 次以上の項を無視することで全微分が求まります。

を得ます。ここで、1行目から2行目への変形で、任意の列ベクトル $a, b \in \mathbb{R}^N$ について、 $ab^\top = ba^\top$ が成り立つことを用いました。

式(3.16)と式(3.23)を比較することで、形式的に勾配が

$$\left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right)^\top = \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^\top} = 2\mathbf{w} \quad (3.24)$$

と求まります。なお、ここでは $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$ と $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^\top}$ の関係をたがいに転置をとったものとして定義しています。これは後に定義する複素勾配を考える際にそのほうが都合がよいかどうです。実ベクトルだけを扱う文献ではこれらを区別しない ($\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^\top}$ とする) こともありますので注意してください。なお、この例の $J(\mathbf{w})$ では勾配を直接求めることは難しくありませんが、関数によっては全微分から求めたほうが簡単なものもあります。その他の例やより詳しい説明は文献3)を参照してください。

3.2 複素関数の微分

3.2.1 正則関数

ここでは複素関数の微分について見ていきましょう。まず、最も簡単な場合として、一つの複素数（スカラー）を引数にもち、複素数値をとる関数 $f(z)$ について考えます。複素微分可能な関数（complex differentiable function, 正則関数（analytic function））はつぎで定義されます¹⁸⁾。

定義 3.3 (正則関数) $D \subseteq \mathbb{C}$ を関数 $f : D \rightarrow \mathbb{C}$ の定義域とする。

$$\lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} \quad (3.25)$$

が任意の内点 $z \in D$ について存在するとき、関数 f は領域 D で複素微分可能（正則）であるという。

複素関数論の基本的な結果として、複素微分可能であるための一般的な条件が存在します。具体的には、関数 f は、**実微分可能** (real differentiable) であり、かつつぎの**コーシー・リーマンの方程式** (Cauchy-Riemann equations) が成り立つとき、またそのときに限り、複素微分可能です⁴⁾。

定義 3.4 (コーシー・リーマンの方程式)

$$\frac{\partial \Re\{f\}}{\partial x} = \frac{\partial \Im\{f\}}{\partial y} \quad (3.26)$$

$$\frac{\partial \Re\{f\}}{\partial y} = -\frac{\partial \Im\{f\}}{\partial x} \quad (3.27)$$

ただし、 $z = x + jy$, $x, y \in \mathbb{R}$ である。

ここで、実微分可能とは、複素数 $z = x + jy$ を引数にもつ複素関数 $f(z)$ を、実数 x, y を引数にもつ実 2 変数関数 $f(x, y)$ と見たときに、それが全微分可能であることをいいます。

コーシー・リーマンの方程式についても少し確認しておきましょう。複素微分可能とは $\Delta f = f(z + \Delta z) - f(z) = K\Delta z$ としたときに、 K が Δz に依存せずに定まるることを意味します。 Δz を実部と虚部に分けて、 $\Delta z = \Delta x + j\Delta y$, $\Delta x, \Delta y \in \mathbb{R}$ とすると

$$\Delta f = K\Delta z = K(\Delta x + j\Delta y) = K\Delta x + jK\Delta y \quad (3.28)$$

となります。一方、 $f(z) = u(x, y) + jv(x, y)$, $u(\cdot), v(\cdot) \in \mathbb{R}$ として、 $\Delta f = \Delta u + j\Delta v$, $\Delta u, \Delta v \in \mathbb{R}$ とすると

$$\begin{aligned} \Delta f &= \frac{\partial u}{\partial x} \Delta x + \frac{\partial u}{\partial y} \Delta y + j \left(\frac{\partial v}{\partial x} \Delta x + \frac{\partial v}{\partial y} \Delta y \right) \\ &= \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} \right) \Delta x + \left(\frac{\partial u}{\partial y} + j \frac{\partial v}{\partial y} \right) \Delta y \end{aligned} \quad (3.29)$$

となります。ここで、 $\Delta x, \Delta y$ に依存せずに K が一意に定まることから

$$j \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} \right) = \frac{\partial u}{\partial y} + j \frac{\partial v}{\partial y} \quad (3.30)$$

すなわち

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad (3.31)$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y} \quad (3.32)$$

が得られます。

3.2.2 正則でない複素関数の例

信号処理の問題では電力や2乗ユークリッド距離の最小化について議論することが多いですが、そのような場合によく現れる関数 $f(z) = |z|^2 = zz^*$ について考えてみましょう。

前述の複素微分の定義式に代入すると

$$\begin{aligned} \lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} &= \lim_{\Delta z \rightarrow 0} \frac{|z + \Delta z|^2 - |z|^2}{\Delta z} \\ &= \lim_{\Delta z \rightarrow 0} \frac{(z + \Delta z)(z + \Delta z)^* - zz^*}{\Delta z} \\ &= \lim_{\Delta z \rightarrow 0} \frac{\Delta zz^* + z(\Delta z)^* + \Delta z(\Delta z)^*}{\Delta z} \end{aligned} \quad (3.33)$$

となります。しかしながら、この式は Δz をどのように0に近づけるかによって値が変わってしまうため、極限が存在しないことがわかります。実際、 $\Delta z = \Delta x + j\Delta y$, $\Delta x, \Delta y \in \mathbb{R}$ として、まず $\Delta x \rightarrow 0$ とするとこの値は $z^* - z - j\Delta y$ となり、さらに $\Delta y \rightarrow 0$ とすると $z^* - z$ となります。一方、先に $\Delta y \rightarrow 0$ とすると $z^* + z + \Delta x$ となり、さらに $\Delta x \rightarrow 0$ とすると $z^* + z$ となります。一般に、 $z^* - z \neq z^* + z$ ですので、これは関数 $f(z) = |z|^2 = zz^*$ が複素微分可能でないことを意味します。 $f(z) = |z|^2 = x^2 + y^2$ は直観的にも滑らかな関数ですが、複素関数は滑らかでも微分可能とは限りません。ではこのような関数 $f(z)$ の勾配を求めるにはどうすればよいのでしょうか。

$f(z)$ は実微分可能ですので、一つの方法としては引数を実部と虚部に分けてそれぞれで偏微分することが考えられます。この方法は正攻法なのですが、対

象となる関数が複雑になると計算が煩雑になってしまうという問題があります。一方、通信のための信号処理の論文や教科書を読むと、このような関数も複素数で微分されています。これはどのような微分を考えているのでしょうか。

3.2.3 ウィルティンガー微分（スカラー引数）

正則ではないが実微分可能な関数の勾配を求める際に有効なのが、 z と z^* を独立な変数として扱って関数 $f(z)$ の全微分を考えるというアプローチです。

引数 $z = x + jy$ をもつ実微分可能関数 $f(z)$ の全微分は、独立な変数 $x, y \in \mathbb{R}$ を用いて

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy \quad (3.34)$$

と書けます。ここで

$$z = x + jy \quad (3.35)$$

$$z^* = x - jy \quad (3.36)$$

より

$$dz = dx + jdy \quad (3.37)$$

$$dz^* = dx - jdy \quad (3.38)$$

となるので

$$dx = \frac{1}{2}(dz + dz^*) \quad (3.39)$$

$$dy = \frac{1}{2j}(dz - dz^*) \quad (3.40)$$

を得ます。これを、式 (3.34) に代入すると

$$\begin{aligned} df &= \frac{\partial f}{\partial x} \frac{dz + dz^*}{2} + \frac{\partial f}{\partial y} \frac{dz - dz^*}{2j} \\ &= \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) dz + \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) dz^* \end{aligned} \quad (3.41)$$

となります。ここで、 f を形式的に独立な変数 z, z^* の関数と見ると、その全微分は

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial z^*} dz^* \quad (3.42)$$

とも書くことができます。式(3.41)と式(3.42)を見比べると、つぎのように定義すれば実数の場合と同様の表現が得られ、都合がよいことがわかります^{18), 19)}。

定義 3.5 (スカラー値複素関数のウィルティンガー微分)

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) \quad (3.43)$$

$$\frac{\partial f}{\partial z^*} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) \quad (3.44)$$

これはウィルティンガー微分と呼ばれます。

簡単な例として、 $z = x + jy$ として $f(z) = z$ と $f(z) = z^*$ の場合を考えてみると

$$\frac{\partial z}{\partial z} = \frac{1}{2} \left(\frac{\partial(x + jy)}{\partial x} - j \frac{\partial(x + jy)}{\partial y} \right) = \frac{1}{2}(1 - j \cdot j) = 1 \quad (3.45)$$

$$\frac{\partial z^*}{\partial z^*} = \frac{1}{2} \left(\frac{\partial(x - jy)}{\partial x} + j \frac{\partial(x - jy)}{\partial y} \right) = \frac{1}{2}(1 + j \cdot (-j)) = 1 \quad (3.46)$$

$$\frac{\partial z}{\partial z^*} = \frac{1}{2} \left(\frac{\partial(x + jy)}{\partial x} + j \frac{\partial(x + jy)}{\partial y} \right) = \frac{1}{2}(1 + j \cdot j) = 0 \quad (3.47)$$

$$\frac{\partial z^*}{\partial z} = \frac{1}{2} \left(\frac{\partial(x - jy)}{\partial x} - j \frac{\partial(x - jy)}{\partial y} \right) = \frac{1}{2}(1 - j \cdot (-j)) = 0 \quad (3.48)$$

となります。これより、ウィルティンガー微分を計算する際に z と z^* を形式的に独立変数として扱ってよいこと、すなわち $\frac{\partial}{\partial z}$ を評価する際には z^* を形式的に定数として z に関する偏微分を計算し、 $\frac{\partial}{\partial z^*}$ を評価する際には z を形式的に定数として z^* に関する偏微分を計算してもよいこと、が確認されます。また、この結果を用いて先ほどの $f(z) = |z|^2$ の場合について考えてみると

$$\frac{\partial|z|^2}{\partial z} = \frac{\partial zz^*}{\partial z} = z^* \quad (3.49)$$

$$\frac{\partial|z|^2}{\partial z^*} = \frac{\partial zz^*}{\partial z^*} = z \quad (3.50)$$

となります。

ウィルティンガー微分の定義を用いると

$$\begin{aligned} \frac{\partial f}{\partial z^*} &= \frac{1}{2} \left(\frac{\partial}{\partial x} (\Re\{f\} + j\Im\{f\}) + j \frac{\partial}{\partial y} (\Re\{f\} + j\Im\{f\}) \right) \\ &= \frac{1}{2} \left(\frac{\partial \Re\{f\}}{\partial x} - \frac{\partial \Im\{f\}}{\partial y} \right) + j \frac{1}{2} \left(\frac{\partial \Re\{f\}}{\partial y} + \frac{\partial \Im\{f\}}{\partial x} \right) \end{aligned} \quad (3.51)$$

となります。この実部と虚部がそれぞれ 0 になるという式からコーシー・リーマンの方程式が得られます。つまり、コーシー・リーマンの方程式は

$$\frac{\partial f}{\partial z^*} = 0 \quad (3.52)$$

と簡潔に書くことができます。これより複素微分可能な関数は z^* に依存しないということがわかります。複素数を引数にもつ実数値関数は、出力として実数値をとるために虚部を打ち消す必要があるため、そのような関数は一般に z だけでなく z^* にも依存します。これが、通信の信号処理の問題で考えるコスト関数（コストの大小を議論する必要があるので、引数は複素数でも関数の値は実数となる）が一般に複素微分可能ではなく、複素領域でその勾配を求めるためにウィルティンガー微分のような手法が必要とされる理由です。

3.2.4 ウィルティンガー微分（ベクトル引数）

つぎに、複素数値をとる関数 f の引数が複素ベクトル $\mathbf{z} = [z_1 \ z_2 \ \cdots \ z_N]^T \in \mathbb{C}^N$ の場合を考えましょう。 $z_n = x_n + jy_n$, $x_n, y_n \in \mathbb{R}$ とすると、関数 $f(\mathbf{z})$ の全微分は

$$df = \sum_{n=1}^N \left(\frac{\partial f}{\partial x_n} dx_n + \frac{\partial f}{\partial y_n} dy_n \right) \quad (3.53)$$

と書けますが、ここで $dz_n = dx_n + jdy_n$ および $dz_n^* = dx_n - jdy_n$ を用いると

$$df = \sum_{n=1}^N \left\{ \frac{1}{2} \left(\frac{\partial f}{\partial x_n} - j \frac{\partial f}{\partial y_n} \right) dz_n + \frac{1}{2} \left(\frac{\partial f}{\partial x_n} + j \frac{\partial f}{\partial y_n} \right) dz_n^* \right\} \quad (3.54)$$

と書けます。一方

$$\begin{aligned} \frac{\partial f}{\partial z} &= \begin{bmatrix} \frac{\partial f}{\partial z_1} & \frac{\partial f}{\partial z_2} & \cdots & \frac{\partial f}{\partial z_N} \end{bmatrix}, \\ dz &= [dz_1 \ dz_2 \ \cdots \ dz_N]^T \end{aligned} \quad (3.55)$$

および

$$\begin{aligned} \frac{\partial f}{\partial z^*} &= \begin{bmatrix} \frac{\partial f}{\partial z_1^*} & \frac{\partial f}{\partial z_2^*} & \cdots & \frac{\partial f}{\partial z_N^*} \end{bmatrix}, \\ dz^* &= [dz_1^* \ dz_2^* \ \cdots \ dz_N^*]^T \end{aligned} \quad (3.56)$$

と定義して、 z, z^* を独立な変数とみなすと、全微分

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial z^*} dz^* \quad (3.57)$$

が得られます。これらを比べることで、複素ベクトルを引数にもつスカラー値の複素関数 f のウィルティンガー微分はつぎで定義されます^{18), 19)}。

定義 3.6 (スカラー値複素関数のウィルティンガー微分 (引数が複素ベクトルの場合))

$$\begin{aligned} \frac{\partial f}{\partial z} &= \begin{bmatrix} \frac{\partial f}{\partial z_1} & \frac{\partial f}{\partial z_2} & \cdots & \frac{\partial f}{\partial z_N} \end{bmatrix} \\ &= \left[\frac{1}{2} \left(\frac{\partial f}{\partial x_1} - j \frac{\partial f}{\partial y_1} \right) \ \cdots \ \frac{1}{2} \left(\frac{\partial f}{\partial x_N} - j \frac{\partial f}{\partial y_N} \right) \right] \quad (3.58) \end{aligned}$$

$$\begin{aligned} \frac{\partial f}{\partial z^*} &= \begin{bmatrix} \frac{\partial f}{\partial z_1^*} & \frac{\partial f}{\partial z_2^*} & \cdots & \frac{\partial f}{\partial z_N^*} \end{bmatrix} \\ &= \left[\frac{1}{2} \left(\frac{\partial f}{\partial x_1} + j \frac{\partial f}{\partial y_1} \right) \ \cdots \ \frac{1}{2} \left(\frac{\partial f}{\partial x_N} + j \frac{\partial f}{\partial y_N} \right) \right] \quad (3.59) \end{aligned}$$

全微分 df は

$$df = \begin{bmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial z^*} \end{bmatrix} \begin{bmatrix} dz \\ dz^* \end{bmatrix} \quad (3.60)$$

と書けるので、複素ベクトルを引数にもつスカラー値複素関数 f の勾配は

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial z^*} \end{bmatrix} \quad (3.61)$$

で定義されます。なお、本書では実関数の場合と同様に $\frac{\partial f}{\partial z}$, $\frac{\partial f}{\partial z^*}$ を行ベクトルで定義し

$$\frac{\partial f}{\partial z^H} = \left(\frac{\partial f}{\partial z^*} \right)^T \quad (3.62)$$

とします。ただし、この定義も文献ごとに流儀があるので注意しましょう。

3.2.5 複素勾配

信号処理の教科書では多くの場合、複素関数 $f(z)$ の勾配ベクトルが

$$\nabla_e f = \begin{bmatrix} \frac{\partial f}{\partial x_1} + j \frac{\partial f}{\partial y_1} \\ \frac{\partial f}{\partial x_2} + j \frac{\partial f}{\partial y_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} + j \frac{\partial f}{\partial y_N} \end{bmatrix} \quad (3.63)$$

で定義されます¹²⁾。これは、本書の勾配の表記を用いると

$$\nabla_e f = 2 \frac{\partial f}{\partial z^H} \quad (3.64)$$

と書き直せますが、一般の複素関数 $f(z)$ の勾配の式 (3.61) とは異なっています。特に、勾配が $\mathbf{0}$ になるパラメータを求めるという点では、式 (3.63) が $\mathbf{0}$ になるようにしても、式 (3.61) に含まれる $\frac{\partial f}{\partial z}$ が $\mathbf{0}$ になるとは限りません。これで問題ないのでしょうか。

一般に、式(3.57)より

$$df^* = \frac{\partial f^*}{\partial z} dz + \frac{\partial f^*}{\partial z^*} dz^* \quad (3.65)$$

$$(df)^* = \left(\frac{\partial f}{\partial z} \right)^* dz^* + \left(\frac{\partial f}{\partial z^*} \right)^* dz \quad (3.66)$$

となります。微分と複素共役をとる演算は交換可能なので $df^* = (df)^*$ となることから

$$\frac{\partial f^*}{\partial z^*} = \left(\frac{\partial f}{\partial z} \right)^* \quad (3.67)$$

$$\frac{\partial f}{\partial z^*} = \left(\frac{\partial f^*}{\partial z} \right)^* \quad (3.68)$$

が成り立ちます。

ここで、関数 f が実数値をとる特別な場合には、 $f = f^*$ となるので

$$\frac{\partial f}{\partial z^*} = \left(\frac{\partial f}{\partial z} \right)^* \quad (3.69)$$

となり、複素勾配は

$$\nabla f = \left[\left(\frac{\partial f}{\partial z^*} \right)^* \quad \frac{\partial f}{\partial z^*} \right] \quad (3.70)$$

で与えられます。これより、関数 f が実数値をとる場合は、複素勾配が $\nabla f = 0$ となることと、 $\frac{\partial f}{\partial z} = 0$, $\frac{\partial f}{\partial z^*} = 0$ がいずれも同値であり、式(3.63)による勾配の定義の正当性、すなわち、 $\nabla f = 0$ となるパラメータを求める際に $\frac{\partial f}{\partial z^*} = 0$ となるパラメータを求めるこの正当性、が確認されます。関数 f が複素数値をとる場合には一般に式(3.69)が成り立たず、式(3.63)による勾配の定義では不十分であることに注意します。

3.2.6 ウィルティンガー微分の具体例

最後に、ウィルティンガー微分の具体例をいくつか記します。典型的な通信の信号処理に現れるウィルティンガー微分はそれほど種類が多くなく、ここに

書かれている例を計算ルールとして覚えておくだけで、ほとんどの場面で簡単に勾配が計算できます。

計算ルール 3.1 (引数が複素ベクトルの場合) a, A をそれぞれ定係数の列ベクトル、および行列とすると以下のようになる。

$$\frac{\partial}{\partial z^H} (z^H a) = a \quad (3.71)$$

$$\frac{\partial}{\partial z^H} (a^H z) = 0 \quad (3.72)$$

$$\frac{\partial}{\partial z^H} (z^H A z) = A z \quad (3.73)$$

これは実ベクトルによる実関数の微分の場合と異なっていることに注意します。参考までに x が実数の列ベクトルの場合は、つぎのようになります²⁰⁾。

計算ルール 3.2 (引数が実ベクトルの場合) a, A をそれぞれ定係数の列ベクトル、および行列とすると以下のようになる。

$$\frac{\partial}{\partial x^T} (x^T a) = a \quad (3.74)$$

$$\frac{\partial}{\partial x^T} (a^T x) = a \quad (3.75)$$

$$\frac{\partial}{\partial x^T} (x^T A x) = Ax + A^T x \quad (3.76)$$

ただし、 A が相関行列のように対称行列の場合は

$$\frac{\partial}{\partial x^T} (x^T A x) = 2Ax \quad (3.77)$$

です。

本書では行列を引数にもつ関数のウィルティンガー微分については触れませんでしたが、引数 Z が複素行列の場合も計算ルールとして覚えておくとよいです^{18), 19)}。

計算ルール 3.3 (引数が複素行列の場合) A を定係数の行列とすると以下のようにになる。

$$\frac{\partial}{\partial Z^H} (\text{tr}\{Z^H A\}) = A \quad (3.78)$$

$$\frac{\partial}{\partial Z^H} (\text{tr}\{A^H Z\}) = 0 \quad (3.79)$$

$$\frac{\partial}{\partial Z^H} (\text{tr}\{Z^H A Z\}) = AZ \quad (3.80)$$

3.3 む す び

本章では、最適なパラメータなどを求める際に必要となる多変数関数の微分についておさらいしました。特に、通信の信号処理では複素数の引数をもつ実数値関数を扱うことが多いため、そのような関数の複素勾配を容易に計算可能な手法であるウィルティンガー微分について説明しました。数学的な内容で嫌だなと感じられたかもしれません、ウィルティンガー微分は確率的勾配降下法 (stochastic gradient descent, SGD) など機械学習でも大活躍しますので、ぜひマスターしておいてください。ウィルティンガー微分や複素信号の解析に関する説明は、以前は文献21)などの一部の解析学の教科書や文献12), 22)などの信号処理の教科書の付録にしか見られませんでしたが、最近では文献18), 19)などの優れた文献があります。本書では詳しく説明しなかった、引数が行列の場合のウィルティンガー微分やベクトル値、行列値をとる複素関数のウィルティンガー微分などについてはこれらの文献を参照してください。実数のベクトル、行列による微分については、文献20)も工学者向けに書かれていてお勧めです。

4

線形逆問題のための 基本的な手法

[Next SIP](#)

通信システムのための信号処理の問題は、通信路応答推定や通信路等化、MIMO信号検出、マルチユーザ信号検出など、そのほとんどが線形観測モデルにおける逆問題（inverse problem）として定式化されます。本章では、線形観測モデルを導入した後、線形逆問題を解くためのいくつかの基本的な手法を説明します。

4.1 線形観測モデル

本章では、興味のある未知ベクトル $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^T \in \mathbb{C}^N$ を、既知の行列 $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_N] \in \mathbb{C}^{M \times N}$ を用いて観測したときに、得られた観測ベクトル $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_M]^T \in \mathbb{C}^M$ から \mathbf{x} を推定する問題を考えます。つまり

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v} \quad (4.1)$$

において \mathbf{y} と \mathbf{A} から \mathbf{x} を推定するという問題です。ここで、 $\mathbf{v} = [v_1 \ v_2 \ \cdots \ v_M]^T \in \mathbb{C}^M$ は、平均が $E[\mathbf{v}] = \mathbf{0}$ で相関行列が $\mathbf{R}_v = E[\mathbf{v}\mathbf{v}^H] = \sigma_v^2 \mathbf{I}_M$ で与えられる加法性白色雑音ベクトル（additive white noise vector）とします。未知ベクトル \mathbf{x} を直接観測することができればよいのですが、通信やその他の工学の多くの問題では \mathbf{x} を直接観測することができず、 \mathbf{x} にある行列 \mathbf{A} （観測行列（measurement matrix）といいます）を乗算し、それに観測雑音ベクトル（measurement noise vector）を加算したものが観測できます。このような観測のモデルは、線形観測モデル（linear measurement model）と呼ばれます。

ます。また、線形観測モデルにおいて未知ベクトル \mathbf{x} を推定する問題は**線形逆問題** (linear inverse problem) と呼ばれます。以下では、線形逆問題を解くためのいくつかの基本的な方法について説明します。なお、未知ベクトル \mathbf{x} は推定方法によって確率的に扱われる場合と、確定的に扱われる場合の両方があることに注意しましょう。

式(4.1)の複素信号モデルは、一般にその2倍のサイズの実数信号モデルで表すこともできます。式(4.1)を実部と虚部に分けて書くと

$$\begin{aligned}\Re\{\mathbf{y}\} + j\Im\{\mathbf{y}\} &= (\Re\{\mathbf{A}\} + j\Im\{\mathbf{A}\})(\Re\{\mathbf{x}\} + j\Im\{\mathbf{x}\}) + \Re\{\mathbf{v}\} + j\Im\{\mathbf{v}\} \\ &= (\Re\{\mathbf{A}\}\Re\{\mathbf{x}\} - \Im\{\mathbf{A}\}\Im\{\mathbf{x}\} + \Re\{\mathbf{v}\}) \\ &\quad + j(\Im\{\mathbf{A}\}\Re\{\mathbf{x}\} + \Re\{\mathbf{A}\}\Im\{\mathbf{x}\} + \Im\{\mathbf{v}\})\end{aligned}$$

となるので

$$\bar{\mathbf{y}} = \begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} \in \mathbb{R}^{2M \times 1} \quad (4.2)$$

$$\bar{\mathbf{A}} = \begin{bmatrix} \Re\{\mathbf{A}\} & -\Im\{\mathbf{A}\} \\ \Im\{\mathbf{A}\} & \Re\{\mathbf{A}\} \end{bmatrix} \in \mathbb{R}^{2M \times 2N} \quad (4.3)$$

$$\bar{\mathbf{x}} = \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix} \in \mathbb{R}^{2N \times 1} \quad (4.4)$$

$$\bar{\mathbf{v}} = \begin{bmatrix} \Re\{\mathbf{v}\} \\ \Im\{\mathbf{v}\} \end{bmatrix} \in \mathbb{R}^{2M \times 1} \quad (4.5)$$

と定義することで、複素数の場合の2倍のサイズの実ベクトルを用いて

$$\bar{\mathbf{y}} = \bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{v}} \quad (4.6)$$

と書くことができます。この変換を用いることで、実数で実現されているアルゴリズムで複素数の信号を処理することが可能になります。

4.2 ZF 推定と最小 2 乗推定

4.2.1 ZF 推定

式 (4.1) の線形観測モデルで $M \geq N$ が成り立つとき、未知ベクトル \mathbf{x} に対する最もシンプルな推定法として、**ZF 推定** (zero-forcing estimation) があります。ZF 推定による \mathbf{x} の推定値は、 $N \times M$ の重み行列 \mathbf{W}_{zf}^H を用いて

$$\hat{\mathbf{x}}_{\text{zf}} = \mathbf{W}_{\text{zf}}^H \mathbf{y} \quad (4.7)$$

で得られます。ZF 推定は観測雑音がない場合に誤差なく \mathbf{x} を推定できる手法で、その重み行列 \mathbf{W}_{zf}^H は

$$\mathbf{W}_{\text{zf}}^H \mathbf{A} = \mathbf{I}_N \quad (4.8)$$

を満足します。このように、観測ベクトル \mathbf{y} に対してある行列を乗算することで推定値を得る方法は、一般に**線形推定** (linear estimation) と呼ばれます。ここで、行列 \mathbf{W}_{zf} そのものではなく \mathbf{W}_{zf} のエルミート転置が重み行列として \mathbf{y} に乗算されていますが、これは表記が簡単になるからで、それ以外に特に理由はありません。

式 (4.8) を用いると式 (4.7) は

$$\hat{\mathbf{x}}_{\text{zf}} = \mathbf{x} + \mathbf{W}_{\text{zf}}^H \mathbf{v} \quad (4.9)$$

となり、 $\mathbf{v} = \mathbf{0}$ のときに $\hat{\mathbf{x}}_{\text{zf}} = \mathbf{x}$ となることが確認できます。

\mathbf{A} が正則な正方行列 ($M = N$) の場合には、式 (4.8) の条件から重み行列が

$$\mathbf{W}_{\text{zf}}^H = \mathbf{A}^{-1} \quad (4.10)$$

と一意に定まります。

一方、 \mathbf{A} のサイズが $M > N$ の場合、つまり、観測ベクトル \mathbf{y} の要素数が未知ベクトル \mathbf{x} の要素数よりも多い場合には、一般に式 (4.8) から \mathbf{W}_{zf} が一意に

定まりません。このようなときには、式(4.8)を満足する重み行列のうちで推定値 \hat{x}_{zf} の信号対雑音電力比 (signal-to-noise power ratio, SNR) を最大にするものを \mathbf{W}_{zf} として選ぶことにします。なお、 \mathbf{A} のサイズが $M < N$ の場合には、式(4.8)を満足する \mathbf{W}_{zf} が存在しないので、ZF 推定は実現できません。このような場合には、後述する最小ノルム解や圧縮センシングなどを利用することになります。

\mathbf{A} のサイズが $M > N$ の場合について考えます。式(4.9)において、信号成分は重み行列に依存せず \mathbf{x} そのものであり、また、雑音成分 $\mathbf{W}_{\text{zf}}^H \mathbf{v}$ の電力は

$$\begin{aligned} E[(\mathbf{W}_{\text{zf}}^H \mathbf{v})^H \mathbf{W}_{\text{zf}}^H \mathbf{v}] &= E[\text{tr}\{\mathbf{W}_{\text{zf}}^H \mathbf{v} \mathbf{v}^H \mathbf{W}_{\text{zf}}\}] \\ &= \text{tr}\{\mathbf{W}_{\text{zf}}^H E[\mathbf{v} \mathbf{v}^H] \mathbf{W}_{\text{zf}}\} \\ &= \sigma_v^2 \text{tr}\{\mathbf{W}_{\text{zf}}^H \mathbf{W}_{\text{zf}}\} \end{aligned} \quad (4.11)$$

で与えられることから、SNR を最大にする \mathbf{W}_{zf} は、推定値 \hat{x}_{zf} の雑音成分の電力 $\sigma_v^2 \text{tr}\{\mathbf{W}_{\text{zf}}^H \mathbf{W}_{\text{zf}}\}$ を最小にすることがわかります。つまり

$$\mathbf{W}_{\text{zf}} = \arg \min_{\mathbf{W} \in \mathbb{C}^{M \times N}} \text{tr}\{\mathbf{W}^H \mathbf{W}\}, \quad s.t. \quad \mathbf{W}^H \mathbf{A} = \mathbf{I}_N \quad (4.12)$$

なる最適化問題を解くことで推定値の SNR を最大にする \mathbf{W}_{zf} が得られます。

最適化問題の表記に慣れていない方のために簡単に説明しておきます。 \arg は argument の略で、複素数の偏角ではなく後ろ（右側）に続く関数の引数を抽出する関数を表します。 \min は minimize の略で、 $\min_{\mathbf{W} \in \mathbb{C}^{M \times N}}$ はその右側の式（信号処理ではよくコスト関数と呼ばれます）を、 \mathbf{W} を変化させることで最小化する関数を表します。 $s.t.$ は subject to あるいは such that の略で、その右側に書かれている式が制約条件 (constraint) であることを意味します。まとめると、式(4.12)は「制約条件 $\mathbf{W}^H \mathbf{A} = \mathbf{I}_N$ の下でコスト関数 $\text{tr}\{\mathbf{W}^H \mathbf{W}\}$ の値を \mathbf{W} について最小化し、コスト関数の最小値を与える \mathbf{W} を重み行列 \mathbf{W}_{zf} としなさい」という意味になります。

なお、式(4.11)で雑音成分の電力を計算する際に、行列のトレースの性質

$\text{tr}\{\mathbf{AB}\} = \text{tr}\{\mathbf{BA}\}^\dagger$ を用いて、期待値操作内の行列の乗算の順番を入れ替えていることに注意します。特に、 \mathbf{a} , \mathbf{b} が列ベクトルのとき、 $\mathbf{a}^\text{H}\mathbf{b}$ はスカラーとなり、トレースをとっても変化しないため

$$\mathbf{a}^\text{H}\mathbf{b} = \text{tr}\{\mathbf{a}^\text{H}\mathbf{b}\} = \text{tr}\{\mathbf{b}\mathbf{a}^\text{H}\} \quad (4.13)$$

なる関係があり、これを式(4.11)の1行目で利用しています。これは統計的信号処理で頻繁に用いられるテクニックで、これにより確定的な行列である重み行列 \mathbf{W}_{zf} を期待値操作の外に出し、重み行列についての（ウィルティンガー）微分を行うことが可能になります。

式(4.12)の最適化問題は標準的な制約付き最小化問題 (constrained minimization problem) なので、ラグランジュ未定乗数法 (method of Lagrange multiplier)^{12), 23)} を用いて解くことができます。ラグランジュ関数 (Lagrangian) を

$$\begin{aligned} \mathcal{L}_{\text{zf}}(\mathbf{W}) &= \text{tr}\{\mathbf{W}^\text{H}\mathbf{W}\} + \sum_{n=1}^N \phi_n^\text{H}(\mathbf{W}^\text{H}\mathbf{a}_n - \mathbf{e}_n) \\ &= \text{tr}\{\mathbf{W}^\text{H}\mathbf{W}\} + \sum_{n=1}^N \text{tr}\{(\mathbf{W}^\text{H}\mathbf{a}_n - \mathbf{e}_n)\phi_n^\text{H}\} \end{aligned} \quad (4.14)$$

と定義します。ここで、 ϕ_n はラグランジュ乗数 (Lagrange multiplier) からなる N 次元列ベクトルで、 \mathbf{e}_n は n 番目の成分のみが 1 で他の成分がすべて 0 の N 次元列ベクトルです。 $\mathcal{L}_{\text{zf}}(\mathbf{W})$ を \mathbf{W}^H でウィルティンガー微分すると

$$\begin{aligned} \frac{\partial \mathcal{L}_{\text{zf}}(\mathbf{W})}{\partial \mathbf{W}^\text{H}} &= \mathbf{W} + \sum_{n=1}^N \mathbf{a}_n \phi_n^\text{H} \\ &= \mathbf{W} + \mathbf{A}\Phi^\text{H} \end{aligned} \quad (4.15)$$

となります。ただし、 $\Phi = [\phi_1 \ \phi_2 \ \cdots \ \phi_N]$ です。さらに、 $\frac{\partial \mathcal{L}_{\text{zf}}(\mathbf{W})}{\partial \mathbf{W}^\text{H}} = \mathbf{0}$ を満足する \mathbf{W} を \mathbf{W}_{zf} とすると

$$\mathbf{W}_{\text{zf}} = -\mathbf{A}\Phi^\text{H} \quad (4.16)$$

[†] 行列 \mathbf{A} , \mathbf{B} のサイズが積 \mathbf{AB} , \mathbf{BA} の両方を計算できるものであることが前提です。

となります。これを式 (4.12) の制約式に代入すると

$$-\Phi A^H A = I_N \quad (4.17)$$

を得て、さらに A が列フルランクであると仮定[†]して $A^H A$ の逆行列 (inverse matrix) を両辺の右から乗算すると

$$\Phi = -(A^H A)^{-1} \quad (4.18)$$

を得ます。したがって、最終的に

$$W_{\text{zf}}^H = (A^H A)^{-1} A^H \quad (4.19)$$

を得ます。この式で $M = N$ とすると $W_{\text{zf}}^H = A^{-1}$ となり、式 (4.10) に一致することから、これは $M = N$ の場合にも有効であることがわかります。

上の ZF 推定の重み行列の導出では、ウィルティンガー微分を用いました。複素数の引数をもつコスト関数を、複素数の場合の 2 倍の実数の引数をもつ関数と考えると、通常の実微分で重み行列を求めることが可能ですが、ウィルティンガー微分を用いることで計算が大幅に簡略化されます。

4.2.2 最小 2 乗推定

最小 2 乗推定 (least squares estimation) では、観測 y と推定値 \hat{x}_{ls} に観測行列 A を乗算したものの誤差 $A\hat{x}_{\text{ls}} - y$ の ℓ_2 ノルムが最小になるように推定値を選びます。つまり

$$\hat{x}_{\text{ls}} = \arg \min_{x \in \mathbb{C}^N} \|Ax - y\|_2^2 \quad (4.20)$$

なる最適化問題を解くことで推定値を得ます。この最適化問題のコスト関数は

$$\|Ax - y\|_2^2 = (Ax - y)^H (Ax - y)$$

[†] A がフルランクでないと $A^H A$ の逆行列が存在しないため、このように仮定しています。逆行列はいつでも存在するわけではないことに注意しましょう。

$$= \mathbf{x}^H \mathbf{A}^H \mathbf{A} \mathbf{x} - \mathbf{x}^H \mathbf{A}^H \mathbf{y} - \mathbf{y}^H \mathbf{A} \mathbf{x} + \mathbf{y}^H \mathbf{y}$$

となることから

$$\frac{\partial}{\partial \mathbf{x}^H} \|\mathbf{A} \mathbf{x} - \mathbf{y}\|_2^2 = \mathbf{A}^H \mathbf{A} \mathbf{x} - \mathbf{A}^H \mathbf{y} = 0 \quad (4.21)$$

を解くことで、最小2乗推定値

$$\hat{\mathbf{x}}_{ls} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{y} \quad (4.22)$$

が得られます。これは、ZF推定の式(4.19)と同一のものになっています。つまり、最小2乗推定とZF推定は本質的に同じ推定法であることがわかります。通信の問題では、通信路等化やMIMO信号検出に用いられる場合にはZF推定と呼ばれ、通信路応答推定の場合には最小2乗推定と呼ばれることが多いようです。

4.2.3 雑音強調

ZF推定は**希望信号**(desired signal)成分を忠実に再現することを優先した推定法なので、式(4.12)のようにできるだけ雑音電力を抑えるような重み行列を選択しても、 \mathbf{A} の性質によっては雑音成分の影響で推定値が大きく劣化してしまうことがあります。

ZF推定値に含まれる雑音成分の電力は、式(4.11)に式(4.19)を代入することで

$$\sigma_v^2 \text{tr} \{ \mathbf{W}_{zf}^H \mathbf{W}_{zf} \} = \sigma_v^2 \text{tr} \{ (\mathbf{A}^H \mathbf{A})^{-1} \} \quad (4.23)$$

と求まります。 $\mathbf{A} \in \mathbb{C}^{M \times N}$ が $M \geq N$ で**列フルランク**(full column rank)とし、その**特異値分解**(singular value decomposition, SVD)が

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \Xi \\ \mathbf{0}_{(M-N) \times N} \end{bmatrix} \mathbf{V}^H \quad (4.24)$$

で与えられます。ただし、 $\mathbf{U} \in \mathbb{C}^{M \times M}$, $\mathbf{V} \in \mathbb{C}^{N \times N}$ はユニタリ行列。 Ξ は \mathbf{A} の特異値 (singular value) からなる対角行列 $\Xi = \text{diag}[\xi_1 \ \xi_2 \ \cdots \ \xi_N]$ です。このとき

$$\mathbf{A}^H \mathbf{A} = \mathbf{V} \Xi^2 \mathbf{V}^H \quad (4.25)$$

となるので、式 (4.23) 中の $(\mathbf{A}^H \mathbf{A})^{-1}$ は

$$(\mathbf{A}^H \mathbf{A})^{-1} = \mathbf{V} \Xi^{-2} \mathbf{V}^H \quad (4.26)$$

となります。したがって、雑音電力は

$$\sigma_v^2 \text{tr} \{ \mathbf{W}_{\text{zf}}^H \mathbf{W}_{\text{zf}} \} = \sigma_v^2 \sum_{n=1}^N \frac{1}{\xi_n^2} \quad (4.27)$$

と表現することができます。これより、 \mathbf{A} の特異値 ξ_n に一つでも 0 に近いものがあれば、ZF 推定値に含まれる雑音電力が非常に大きくなってしまうことがわかります。ZF 推定値に含まれる希望信号電力は \mathbf{A} に依存せず一定値なので、これは ZF 推定値の SNR が大きく劣化することを意味します。この問題は ZF 推定の**雑音強調** (noise enhancement) として知られています。

4.3 MMSE 推 定

ZF 推定は観測行列 \mathbf{A} の情報だけで推定のための重み行列が決定できるという利点がありますが、雑音強調による特性劣化という問題があります。未知ベクトルや雑音の相関行列などに関する情報が利用可能な場合には、**最小平均 2 乗誤差推定** (minimum mean-square-error estimation, MMSE 推定) を用いることで雑音強調の影響を低減することができます。

MMSE 推定には、推定値を得る際に非線形演算も許す一般の MMSE 推定と、重み行列の乗算のみで推定値を得る線形 MMSE 推定があります。よく用いられるのは後者ですが、本節ではまず一般の MMSE 推定から説明します。ここ

で説明する一般の MMSE 推定では、必ずしも式 (4.1) の線形観測モデルを仮定する必要がないことに注意します。

4.3.1 一般的 MMSE 推定

一般的 MMSE 推定では、観測ベクトル \mathbf{y} を引数にもつベクトル値の非線形関数 \mathbf{f} を用いて

$$\hat{\mathbf{x}}_{\text{mmse}} = \mathbf{f}(\mathbf{y}) \quad (4.28)$$

によって推定値を与えることにします。ただし、 \mathbf{f} は観測ベクトル \mathbf{y} が与えられた下での**推定誤差** (estimation error) ベクトル $\mathbf{f}(\mathbf{y}) - \mathbf{x}$ の ℓ_2 ノルムの 2 乗の \mathbf{x} に関する期待値 (**平均 2 乗誤差** (mean-square-error))

$$J_{\text{mmse}}[\mathbf{f}] = E \left[\|\mathbf{f}(\mathbf{y}) - \mathbf{x}\|_2^2 \middle| \mathbf{y} \right] \quad (4.29)$$

が最小になるように決定されます。

ここで、 \mathbf{x} を連続確率変数として、 \mathbf{y} が与えられたときの \mathbf{x} の条件付き確率密度関数を $p(\mathbf{x}|\mathbf{y})$ 、 \mathbf{y} が与えられた下での \mathbf{x} の**条件付き期待値** (conditional expectation) を

$$\mathbf{m}_{\mathbf{x}|\mathbf{y}} = E[\mathbf{x}|\mathbf{y}] = \int \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \quad (4.30)$$

と定義すると†

$$\begin{aligned} J_{\text{mmse}}[\mathbf{f}] &= E \left[\|\mathbf{f}(\mathbf{y}) - \mathbf{x}\|_2^2 \middle| \mathbf{y} \right] \\ &= E \left[\|\mathbf{f}(\mathbf{y}) - \mathbf{m}_{\mathbf{x}|\mathbf{y}} + \mathbf{m}_{\mathbf{x}|\mathbf{y}} - \mathbf{x}\|_2^2 \middle| \mathbf{y} \right] \\ &= \|\mathbf{f}(\mathbf{y}) - \mathbf{m}_{\mathbf{x}|\mathbf{y}}\|_2^2 + E \left[\|\mathbf{m}_{\mathbf{x}|\mathbf{y}} - \mathbf{x}\|_2^2 \middle| \mathbf{y} \right] \end{aligned}$$

† \mathbf{x} が離散確率変数のときは、条件付き確率 $P(\mathbf{x}|\mathbf{y})$ を用いて

$$\mathbf{m}_{\mathbf{x}|\mathbf{y}} = E[\mathbf{x}|\mathbf{y}] = \sum \mathbf{x} P(\mathbf{x}|\mathbf{y})$$

とします。

$$\begin{aligned}
& + \{\mathbf{f}(\mathbf{y}) - \mathbf{m}_{x|y}\}^H E[\{\mathbf{m}_{x|y} - \mathbf{x}\} | \mathbf{y}] \\
& + E[\{\mathbf{m}_{x|y} - \mathbf{x}\}^H | \mathbf{y}] \{\mathbf{f}(\mathbf{y}) - \mathbf{m}_{x|y}\} \\
& = \|\mathbf{f}(\mathbf{y}) - \mathbf{m}_{x|y}\|_2^2 + E[\|\mathbf{m}_{x|y} - \mathbf{x}\|_2^2 | \mathbf{y}] \\
& \geq E[\|\mathbf{m}_{x|y} - \mathbf{x}\|_2^2 | \mathbf{y}]
\end{aligned} \tag{4.31}$$

となります。 $\mathbf{f}(\mathbf{y}) = \mathbf{m}_{x|y}$ のとき、最後の行の不等式で等号が成立するので、条件付き期待値の式 (4.30) を引数 \mathbf{y} をもつ関数と見たときに、これが平均 2 乗誤差を最小にする推定値を与える関数 \mathbf{f} であることがわかります。したがって、一般の MMSE 推定は式 (4.30) の条件付き期待値

$$\hat{\mathbf{x}}_{\text{mmse}} = \mathbf{m}_{x|y} \tag{4.32}$$

によって与えられます。

ここで真の \mathbf{x} と MMSE 推定値の誤差ベクトルの期待値が

$$E[\mathbf{x} - \mathbf{m}_{x|y}] = E[\mathbf{x} - E[\mathbf{x}|y]] \tag{4.33}$$

$$= \int \int p(\mathbf{x}, \mathbf{y}) (\mathbf{x} - E[\mathbf{x}|y]) d\mathbf{x} d\mathbf{y} \tag{4.34}$$

$$= \int p(\mathbf{y}) \left(\int p(\mathbf{x}|y) (\mathbf{x} - E[\mathbf{x}|y]) d\mathbf{x} \right) d\mathbf{y} \tag{4.35}$$

$$= \mathbf{0} \tag{4.36}$$

となることを利用すると、誤差ベクトルの各成分の分散の和は

$$E[\|\mathbf{x} - \mathbf{m}_{x|y}\|_2^2] = \int \int p(\mathbf{x}, \mathbf{y}) \|\mathbf{x} - E[\mathbf{x}|y]\|_2^2 d\mathbf{x} d\mathbf{y} \tag{4.37}$$

$$= \int p(\mathbf{y}) \left(\int p(\mathbf{x}|y) \|\mathbf{x} - E[\mathbf{x}|y]\|_2^2 d\mathbf{x} \right) d\mathbf{y} \tag{4.38}$$

$$= \int p(\mathbf{y}) E[\|\mathbf{x} - E[\mathbf{x}|y]\|_2^2 | \mathbf{y}] d\mathbf{y} \tag{4.39}$$

となります。ここで、任意の \mathbf{y} について、 $E \left[\| \mathbf{x} - \mathbf{f}(\mathbf{y}) \|_2^2 \middle| \mathbf{y} \right]$ の最小値は $\mathbf{f}(\mathbf{y}) = E[\mathbf{x}|\mathbf{y}]$ によって与えられることと、 $p(\mathbf{y})$ と $E \left[\| \mathbf{x} - E[\mathbf{x}|\mathbf{y}] \|_2^2 \middle| \mathbf{y} \right]$ が非負であることから、MMSE 推定値は誤差ベクトルの各成分の分散の和を最小にする推定値であることがわかります。この意味で、MMSE 推定値は**最小分散推定値** (minimum variance estimate) とも呼ばれます。

4.3.2 線形 MMSE 推定

つぎに、式 (4.1) の線形観測モデルを仮定し、線形推定を行うという制約の下での MMSE 推定について考えます。

簡単のため、未知ベクトル \mathbf{x} の平均を $\mathbf{0}$ とすると、線形 MMSE 推定は重み行列 $\mathbf{W}_{\text{lmmse}}$ を用いて

$$\hat{\mathbf{x}}_{\text{lmmse}} = \mathbf{W}_{\text{lmmse}}^H \mathbf{y} \quad (4.40)$$

で与えられます。ここで、 $\mathbf{W}_{\text{lmmse}}$ はつぎの最適化問題によって決定されます。

$$\mathbf{W}_{\text{lmmse}} = \arg \min_{\mathbf{W} \in \mathbb{C}^{M \times N}} E \left[\| \mathbf{W}^H (\mathbf{A}\mathbf{x} + \mathbf{v}) - \mathbf{x} \|_2^2 \right] \quad (4.41)$$

この最適化問題のコスト関数 $J_{\text{lmmse}}(\mathbf{W})$ は

$$\begin{aligned} J_{\text{lmmse}}(\mathbf{W}) &= E \left[\| \mathbf{W}^H (\mathbf{A}\mathbf{x} + \mathbf{v}) - \mathbf{x} \|_2^2 \right] \\ &= E \left[(\mathbf{W}^H \mathbf{A}\mathbf{x} + \mathbf{W}^H \mathbf{v} - \mathbf{x})^H (\mathbf{W}^H \mathbf{A}\mathbf{x} + \mathbf{W}^H \mathbf{v} - \mathbf{x}) \right] \\ &= E \left[\text{tr} \{ (\mathbf{W}^H \mathbf{A}\mathbf{x} + \mathbf{W}^H \mathbf{v} - \mathbf{x})(\mathbf{W}^H \mathbf{A}\mathbf{x} + \mathbf{W}^H \mathbf{v} - \mathbf{x})^H \} \right] \\ &= \text{tr} \{ \mathbf{W}^H \mathbf{A} E[\mathbf{x}\mathbf{x}^H] \mathbf{A}^H \mathbf{W} \} + \text{tr} \{ \mathbf{W}^H \mathbf{A} E[\mathbf{x}\mathbf{v}^H] \mathbf{W} \} \\ &\quad - \text{tr} \{ \mathbf{W}^H \mathbf{A} E[\mathbf{x}\mathbf{x}^H] \} + \text{tr} \{ \mathbf{W}^H E[\mathbf{v}\mathbf{x}^H] \mathbf{A}^H \mathbf{W} \} \\ &\quad + \text{tr} \{ \mathbf{W}^H E[\mathbf{v}\mathbf{v}^H] \mathbf{W} \} - \text{tr} \{ \mathbf{W}^H E[\mathbf{v}\mathbf{x}^H] \} \\ &\quad - \text{tr} \{ E[\mathbf{x}\mathbf{x}^H] \mathbf{A}^H \mathbf{W} \} - \text{tr} \{ E[\mathbf{x}\mathbf{v}^H] \mathbf{W} \} + \text{tr} \{ E[\mathbf{x}\mathbf{x}^H] \} \\ &= \text{tr} \{ \mathbf{W}^H \mathbf{A} \mathbf{R}_x \mathbf{A}^H \mathbf{W} \} - \text{tr} \{ \mathbf{W}^H \mathbf{A} \mathbf{R}_x \} + \sigma_v^2 \text{tr} \{ \mathbf{W}^H \mathbf{W} \} \\ &\quad - \text{tr} \{ \mathbf{R}_x \mathbf{A}^H \mathbf{W} \} + \text{tr} \{ \mathbf{R}_x \} \end{aligned} \quad (4.42)$$

と計算することができます。ただし、 \mathbf{R}_x は未知ベクトル \mathbf{x} の相関行列 $\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^H]$ です。また、最後の行への変形では、信号と雑音の無相関性 $E[\mathbf{x}\mathbf{v}^H] = \mathbf{0}$, $E[\mathbf{v}\mathbf{x}^H] = \mathbf{0}$ を用いています。

コスト関数 $J_{\text{lmmse}}(\mathbf{W})$ を \mathbf{W}^H についてウィルティンガー微分し、それをイコール $\mathbf{0}$ として得られる方程式、すなわち

$$\frac{\partial J_{\text{lmmse}}(\mathbf{W})}{\partial \mathbf{W}^H} = \mathbf{A}\mathbf{R}_x\mathbf{A}^H\mathbf{W} - \mathbf{A}\mathbf{R}_x + \sigma_v^2\mathbf{W} = \mathbf{0} \quad (4.43)$$

を解くことで、線形 MMSE 推定の重み行列

$$\mathbf{W}_{\text{lmmse}}^H = \mathbf{R}_x\mathbf{A}^H (\mathbf{A}\mathbf{R}_x\mathbf{A}^H + \sigma_v^2\mathbf{I}_M)^{-1} \quad (4.44)$$

を得ます。

一般に線形 MMSE 推定値は最小分散推定値（一般的 MMSE 推定値）とは一致しませんが、 (\mathbf{x}, \mathbf{y}) の結合分布がガウス分布（Gaussian distribution）の場合にはこれらが一致します²⁴⁾。

未知ベクトル \mathbf{x} が白色で相関行列が $\mathbf{R}_x = \sigma_x^2\mathbf{I}_N$ となるときは

$$\mathbf{W}_{\text{lmmse}}^H = \mathbf{A}^H \left(\mathbf{A}\mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2}\mathbf{I}_M \right)^{-1} \quad (4.45)$$

となります。ここで、逆行列補題（matrix inversion lemma、付録 A.1 参照）を用いると

$$\begin{aligned} & \left(\mathbf{A}\mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2}\mathbf{I}_M \right)^{-1} \\ &= \frac{\sigma_x^2}{\sigma_v^2}\mathbf{I}_M - \frac{\sigma_x^2}{\sigma_v^2}\mathbf{I}_M \cdot \mathbf{A} \left(\mathbf{I}_N + \mathbf{A}^H \frac{\sigma_x^2}{\sigma_v^2}\mathbf{I}_M \mathbf{A} \right)^{-1} \mathbf{A}^H \cdot \frac{\sigma_x^2}{\sigma_v^2}\mathbf{I}_M \\ &= \frac{\sigma_x^2}{\sigma_v^2} \left\{ \mathbf{I}_M - \frac{\sigma_x^2}{\sigma_v^2} \mathbf{A} \left(\mathbf{I}_N + \frac{\sigma_x^2}{\sigma_v^2} \mathbf{A}^H \mathbf{A} \right)^{-1} \mathbf{A}^H \right\} \\ &= \frac{\sigma_x^2}{\sigma_v^2} \left\{ \mathbf{I}_M - \mathbf{A} \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \mathbf{A}^H \right\} \end{aligned}$$

となるので

$$\begin{aligned}
\mathbf{W}_{\text{lmmse}}^H &= \mathbf{A}^H \frac{\sigma_x^2}{\sigma_v^2} \left\{ \mathbf{I}_M - \mathbf{A} \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \mathbf{A}^H \right\} \\
&= \frac{\sigma_x^2}{\sigma_v^2} \left\{ \mathbf{A}^H - \mathbf{A}^H \mathbf{A} \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \mathbf{A}^H \right\} \\
&= \frac{\sigma_x^2}{\sigma_v^2} \left\{ \mathbf{I}_N - \mathbf{A}^H \mathbf{A} \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \right\} \mathbf{A}^H \\
&= \frac{\sigma_x^2}{\sigma_v^2} \left\{ \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right) \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \right. \\
&\quad \left. - \mathbf{A}^H \mathbf{A} \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \right\} \mathbf{A}^H \\
&= \frac{\sigma_x^2}{\sigma_v^2} \left\{ \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N - \mathbf{A}^H \mathbf{A} \right) \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \right\} \mathbf{A}^H \\
&= \left(\mathbf{A}^H \mathbf{A} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N \right)^{-1} \mathbf{A}^H
\end{aligned} \tag{4.46}$$

と書くこともできます。 $\frac{\sigma_v^2}{\sigma_x^2}$ は送信信号の各成分の電力と観測雑音の各成分の電力の比の逆数、すなわち SNR の逆数になっていることに注意すると、SNR が大きい場合には $\frac{\sigma_v^2}{\sigma_x^2}$ の値が小さいので $\frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_N$ の項を無視すると $\mathbf{W}_{\text{lmmse}}^H \approx (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$ となります。これは、SNR が大きいときには、線形 MMSE 推定の重み行列が ZF 推定の重み行列とほぼ等しくなることを意味します。

4.3.3 線形 MMSE 推定の SINR

ここでは未知ベクトル \mathbf{x} が白色で相関行列が $\mathbf{R}_x = \sigma_x^2 \mathbf{I}_N$ となるときの、線形 MMSE 推定値の信号対干渉プラス雑音電力比 (signal-to-interference-plus-noise power ratio, SINR) について調べてみましょう。ZF 推定では推定値に希望信号成分と雑音成分しか含まれていないため SNR について考えましたが、MMSE 推定では推定値に干渉信号成分も含まれるため、ここでは SNR ではなく SINR について考えることに注意します。

線形 MMSE 推定値 $\hat{\mathbf{x}}_{\text{lmmse}} = [\hat{x}_{\text{lmmse},1} \ \hat{x}_{\text{lmmse},2} \ \cdots \ \hat{x}_{\text{lmmse},N}]^T$ は重み行

列 $\mathbf{W}_{\text{lmmse}} = [\mathbf{w}_{\text{lmmse},1} \ \mathbf{w}_{\text{lmmse},2} \ \cdots \ \mathbf{w}_{\text{lmmse},N}]$ を用いて

$$\hat{x}_{\text{lmmse}} = \mathbf{W}_{\text{lmmse}}^H \mathbf{y} = \begin{bmatrix} \mathbf{w}_{\text{lmmse},1}^H \mathbf{y} \\ \mathbf{w}_{\text{lmmse},2}^H \mathbf{y} \\ \vdots \\ \mathbf{w}_{\text{lmmse},M}^H \mathbf{y} \end{bmatrix} \quad (4.47)$$


と書けます。ただし、 $\mathbf{w}_{\text{lmmse},n}$ は $\mathbf{W}_{\text{lmmse}}$ の n 番目の列ベクトルです。これより、 \mathbf{x} の n 番目の成分の推定値は

$$\begin{aligned} \hat{x}_{\text{lmmse},n} &= \mathbf{w}_{\text{lmmse},n}^H \mathbf{y} \\ &= \mathbf{w}_{\text{lmmse},n}^H \mathbf{A} \mathbf{x} + \mathbf{w}_{\text{lmmse},n}^H \mathbf{v} \\ &= \mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n x_n + \mathbf{w}_{\text{lmmse},n}^H \sum_{i \neq n} \mathbf{a}_i x_i + \mathbf{w}_{\text{lmmse},n}^H \mathbf{v} \end{aligned} \quad (4.48)$$

となります。ここで、3 行目の右辺第 1 項は希望信号の成分、第 2 項は干渉信号成分、第 3 項は雑音成分です。よって、SINR の信号電力は

$$\begin{aligned} E[|\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n x_n|^2] &= |\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n|^2 E[|x_n|^2] \\ &= \sigma_x^2 |\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n|^2 \end{aligned} \quad (4.49)$$

となります。

一方、干渉信号と雑音を足したものの電力は、 \mathbf{x} の n 番目の成分の推定値 $\hat{x}_{\text{lmmse},n}$ 全体の電力から、いま求めた信号成分の電力を引き算することで求められます。

$\hat{x}_{\text{lmmse},n}$ の電力は

$$\begin{aligned} E[|\hat{x}_{\text{lmmse},n}|^2] &= E[|\mathbf{w}_{\text{lmmse},n}^H \mathbf{y}|^2] \\ &= \mathbf{w}_{\text{lmmse},n}^H E[\mathbf{y} \mathbf{y}^H] \mathbf{w}_{\text{lmmse},n} \\ &= \mathbf{w}_{\text{lmmse},n}^H E[(\mathbf{A} \mathbf{x} + \mathbf{v})(\mathbf{A} \mathbf{x} + \mathbf{v})^H] \mathbf{w}_{\text{lmmse},n} \\ &= \sigma_x^2 \mathbf{w}_{\text{lmmse},n}^H \left(\mathbf{A} \mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_M \right) \mathbf{w}_{\text{lmmse},n} \end{aligned} \quad (4.50)$$

となります。これは行列

$$\begin{aligned} & \sigma_x^2 \mathbf{W}_{\text{lmmse}}^H \left(\mathbf{A} \mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_M \right) \mathbf{W}_{\text{lmmse}} \\ &= \sigma_x^2 \mathbf{W}_{\text{lmmse}}^H \left(\mathbf{A} \mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_M \right) \left(\mathbf{A} \mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_M \right)^{-1} \mathbf{A} \\ &= \sigma_x^2 \mathbf{W}_{\text{lmmse}}^H \mathbf{A} \end{aligned} \quad (4.51)$$

の n 番目の対角成分なので

$$E[|\hat{x}_{\text{lmmse},n}|^2] = \sigma_x^2 \mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n \quad (4.52)$$

となります。ここで、右辺の $\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n$ は一見すると複素数に見えますが、左辺の $E[|\hat{x}_{\text{lmmse},n}|^2]$ は電力なので非負の実数であり、したがって、 $\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n$ も非負の実数であることに注意します。

以上より、 \mathbf{x} の n 番目の成分の線形 MMSE 推定値の SINR γ_n は

$$\begin{aligned} \gamma_n &= \frac{|\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n|^2}{\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n - |\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n|^2} \\ &= \frac{\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n}{1 - \mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n} \end{aligned} \quad (4.53)$$

となります。

さらに、SINR γ_n を $\mathbf{w}_{\text{lmmse},n}$ を用いない形で表現してみましょう。先ほど、行列 $\mathbf{W}_{\text{lmmse}}^H \mathbf{A}$ の n 番目の対角成分が $\mathbf{w}_{\text{lmmse},n}^H \mathbf{a}_n$ であることを利用しましたが、これより γ_n の分子は

$$\mathbf{W}_{\text{lmmse}}^H \mathbf{A} = \mathbf{A}^H \left(\mathbf{A} \mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_M \right)^{-1} \mathbf{A} \quad (4.54)$$

の n 番目の対角成分で与えられます。同様に、 γ_n の分母は

$$\mathbf{I}_N - \mathbf{W}_{\text{lmmse}}^H \mathbf{A} = \mathbf{I}_N - \mathbf{A}^H \left(\mathbf{A} \mathbf{A}^H + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I}_M \right)^{-1} \mathbf{A} \quad (4.55)$$

の n 番目の対角成分で与えられます。ここで、また逆行列補題を用いると

$$\begin{aligned}
 \left(A^H A + \frac{\sigma_v^2}{\sigma_x^2} I_N \right)^{-1} &= \frac{\sigma_x^2}{\sigma_v^2} I_N - \frac{\sigma_x^2}{\sigma_v^2} A^H \left(I_M + \frac{\sigma_x^2}{\sigma_v^2} A A^H \right)^{-1} A \frac{\sigma_x^2}{\sigma_v^2} I_N \\
 &= \frac{\sigma_x^2}{\sigma_v^2} \left\{ I_N - A^H \left(A A^H + \frac{\sigma_v^2}{\sigma_x^2} I_M \right)^{-1} A \right\}
 \end{aligned} \tag{4.56}$$

となり

$$W_{\text{lmmse}}^H A = I_N - \frac{\sigma_v^2}{\sigma_x^2} \left(A^H A + \frac{\sigma_v^2}{\sigma_x^2} I_N \right)^{-1} \tag{4.57}$$

$$I_N - W_{\text{lmmse}}^H A = \frac{\sigma_v^2}{\sigma_x^2} \left(A^H A + \frac{\sigma_v^2}{\sigma_x^2} I_N \right)^{-1} \tag{4.58}$$

となるので、 γ_n はより簡単な形として

$$\gamma_n = \frac{\left\{ I_N - \frac{\sigma_v^2}{\sigma_x^2} \left(A^H A + \frac{\sigma_v^2}{\sigma_x^2} I_N \right)^{-1} \right\}_{n,n}}{\left\{ \frac{\sigma_v^2}{\sigma_x^2} \left(A^H A + \frac{\sigma_v^2}{\sigma_x^2} I_N \right)^{-1} \right\}_{n,n}} \tag{4.59}$$

で与えられます。ここで、 $\{\cdot\}_{n,n}$ は行列の n 番目の対角成分を表します。

4.4 減算型干渉除去

これまでに述べた線形の信号推定法は、観測 y に何らかの行列を一度乗算することで x の推定値を得るものでした。式 (4.1) の線形観測モデルでは y の各成分は x の各成分と観測雑音の線形和として与えられるので、 y から x のある特定の成分を推定することを考えると、観測雑音だけでなく x のその成分以外のすべての成分も不要なもの（干渉）となります。このため、線形推定などの何らかの方法で x の推定値が得られたとき、その推定値から y に含まれる干渉成分のレプリカを生成し、 y から干渉成分を除去した信号を利用して再度 x の推定値を求めることで、より高精度の推定値が得られる可能性があります。このようなアイデアに基づいて繰り返し処理によって信号検出を行う方法

が減算型干渉除去（subtractive interference cancellation）と呼ばれる方法です。このような手法は情報理論の分野では 1970 年代から議論されていましたが^{25), 26)}、通信の信号処理の文脈で広く知られるようになったのは符号分割多元接続（code division multiple access, CDMA）のマルチユーザ検出法²⁷⁾（multiuser detection method）に利用されて以降です。

減算型干渉除去は大きく二つのアプローチに分けられ、それぞれ逐次干渉除去（successive interference cancellation, SIC）、並列干渉除去（parallel interference cancellation, PIC）と呼ばれます。以下ではそれぞれの手法について説明します。

4.4.1 逐次干渉除去

SIC は観測ベクトル \mathbf{y} 中の未知ベクトル \mathbf{x} の各成分の SNR が成分ごとに大きく異なる場合に効果的な信号検出法で、観測信号中の SNR が大きい \mathbf{x} の成分から順に検出を行う点に特徴があります。

簡単のため観測ベクトル \mathbf{y} 中の \mathbf{x} の各成分 x_1, x_2, \dots, x_N の SNR がこの順に小さくなっていくとすると、まず、 x_1 を希望信号成分、 x_2, x_3, \dots, x_N をすべて雑音とみなします。すなわち

$$\begin{aligned} \mathbf{y}_{\text{sic}}^{(1)} &= \mathbf{y} \\ &= \mathbf{A}\mathbf{x} + \mathbf{v} \\ &= \mathbf{a}_1x_1 + \left(\sum_{i=2}^N \mathbf{a}_i x_i + \mathbf{v} \right) \end{aligned} \tag{4.60}$$

として 3 行目の右辺第 1 項を希望信号、第 2 項を雑音とみなします。そして、例えば線形推定を行うとすると、 x_1 の推定値 $\hat{x}_{\text{sic},1}$ は

$$\hat{x}_{\text{sic},1} = \mathbf{w}_1^H \mathbf{y}_{\text{sic}}^{(1)} \tag{4.61}$$

と書くことができます。ここで、 $\mathbf{w}_1^H \in \mathbb{C}^M$ は線形推定のための重みベクトルで、式 (4.19) の ZF 推定の重み行列や式 (4.44) の線形 MMSE 推定の重み行列

などの線形推定のための重み行列の 1 番目の行ベクトルが対応します。

つぎに、推定値 $\hat{x}_{\text{sic},1}$ が真の x_1 に一致していると仮定し、 \mathbf{y} から x_1 の成分を除去した観測ベクトル $\mathbf{y}_{\text{sic}}^{(2)}$ を生成します。具体的には

$$\begin{aligned}\mathbf{y}_{\text{sic}}^{(2)} &= \mathbf{y} - \mathbf{a}_1 \hat{x}_{\text{sic},1} \\ &= \sum_{i=2}^N \mathbf{a}_i x_i + \mathbf{v} + \mathbf{a}_1(x_1 - \hat{x}_{\text{sic},1}) \\ &\approx \sum_{i=2}^N \mathbf{a}_i x_i + \mathbf{v}\end{aligned}\quad (4.62)$$

とします。ここで、実際に $\hat{x}_{\text{sic},1} = x_1$ であれば $\mathbf{y}_{\text{sic}}^{(2)}$ は 3 行目の右辺に等しくなり、 \mathbf{y} から直接 x_2 を推定するよりも $\mathbf{y}_{\text{sic}}^{(2)}$ から推定するほうが x_1 による干渉がなくなった分だけ特性が改善します。 x_2 を $\mathbf{y}_{\text{sic}}^{(2)}$ から線形推定する場合には

$$\mathbf{y}_{\text{sic}}^{(2)} \approx \mathbf{a}_2 x_2 + \left(\sum_{i=3}^N \mathbf{a}_i x_i + \mathbf{v} \right) \quad (4.63)$$

と考えて、線形推定の重みベクトル $\mathbf{w}_2^H \in \mathbb{C}^M$ を用いて

$$\hat{x}_{\text{sic},2} = \mathbf{w}_2^H \mathbf{y}_{\text{sic}}^{(2)} \quad (4.64)$$

とすることで推定値を得ます。ここで、 \mathbf{w}_2^H は式 (4.19) や式 (4.44) の 2 番目の行ベクトルとは一般に異なることに注意します。

以上の処理を繰り返して x_1 から x_N までの推定値を求めるのが SIC の基本的な手順です。観測ベクトル中の SNR が最も大きな信号成分にとっては線形推定などの基本的な推定法に比べてメリットはありませんが、観測ベクトル中の SNR が小さい信号ほどメリットが大きいという点が重要です。

ここでは最も基本的な SIC の手法として、未知ベクトル \mathbf{x} の成分ごとの線形推定値をそのまま用いて干渉成分を除去する方法について説明しましたが、未知ベクトル \mathbf{x} の各成分が既知の離散値のいずれかをとることがわかっている場合には、硬判定値を利用して干渉成分を除去することも可能です。例えば、式

(4.62) の 1 行目において、 $\hat{x}_{\text{sic},1}$ をそのまま用いるのではなく $\hat{x}_{\text{sic},1}$ を x_1 のとり得る最も近い離散値に丸めた値 $\check{x}_{\text{sic},1}$ （これを**硬判定値**（hard decision value）といいます）を用いて

$$\mathbf{y}_{\text{sic}}^{(2)} = \mathbf{y} - \mathbf{a}_1 \check{x}_{\text{sic},1} \quad (4.65)$$

とします。観測行列 \mathbf{A} についての正確な情報が利用できる場合には、硬判定値を用いることで推定精度を改善できることが知られています²⁷⁾。

SIC は準最適な信号検出法ですが、ガウス放送通信路（Gaussian broadcast channel, より一般には劣化型放送通信路（degraded broadcast channel））において重畠符号化（superposition coding）と組み合わせて SIC を用いることで通信路容量領域（channel capacity region）の境界を達成できることが知られており、情報理論の文脈でも重要な信号推定法です²⁶⁾。

4.4.2 並列干渉除去

SIC が \mathbf{x} の成分ごとの推定値を用いてその干渉成分を観測ベクトルから削除して他の成分の推定値を逐次的に求める方法であるのに対し、PIC は未知ベクトル \mathbf{x} 全体の仮推定値を用いて観測ベクトルから干渉成分を除去して推定精度を改善していく方法です。

具体的には、線形推定など何らかの方法を用いて観測ベクトル $\mathbf{y}_{\text{pic}}^{(1)} = \mathbf{y}$ から \mathbf{x} の仮推定値 $\hat{\mathbf{x}}_{\text{pic}}^{(1)}$ が得られたとすると、これを用いて

$$\mathbf{y}_{\text{pic}}^{(2)} = \mathbf{y} - \mathbf{A}_{\text{offdiag}} \hat{\mathbf{x}}_{\text{pic}}^{(1)} \quad (4.66)$$

とすることで未知ベクトル \mathbf{x} 間の干渉成分が抑圧された観測ベクトル $\mathbf{y}_{\text{pic}}^{(2)}$ を得ます。ただし、 $\mathbf{A}_{\text{offdiag}}$ は観測行列 \mathbf{A} の主対角成分以外の成分からなる行列です。

ここで、観測行列 \mathbf{A} の主対角成分のみからなる対角行列を \mathbf{A}_{diag} とすると、 $\mathbf{A}_{\text{offdiag}} = \mathbf{A} - \mathbf{A}_{\text{diag}}$ となるので、 $\mathbf{x} \approx \hat{\mathbf{x}}_{\text{pic}}^{(1)}$ と仮定すると

$$\mathbf{y}_{\text{pic}}^{(2)} = \mathbf{Ax} - \mathbf{A}_{\text{offdiag}} \hat{\mathbf{x}}_{\text{pic}}^{(1)} + \mathbf{v}$$

$$\approx A_{\text{diag}}x + v \quad (4.67)$$

と書くことができます。そこで、さらに $y_{\text{pic}}^{(2)}$ から線形推定など何らかの方法で推定値 $\hat{x}_{\text{pic}}^{(2)}$ を求めると、これを用いて上記の式と同様の方法で干渉成分を抑圧することができます。この処理を繰り返すことで、推定精度の向上を図るのが PIC の基本的な考え方です。PIC は SIC と異なり、 x の成分間で観測ベクトル中の SNR の違いがあまりない場合に有効であることが知られています。

4.5 信 号 合 成

同一の信号に対する複数の観測が得られたとき、これらの観測をどのように利用（合成）するとよいかという問題があります。ここではまず、式 (4.1) の線形観測モデルの特別な場合として

$$y = ax + v \quad (4.68)$$

を考えます。ただし、 $x \in \mathbb{C}$, $a = [a_1 \ a_2 \ \dots \ a_M]^T \in \mathbb{C}^M$, $y = [y_1 \ y_2 \ \dots \ y_M]^T \in \mathbb{C}^M$, $v = [v_1 \ v_2 \ \dots \ v_M]^T \in \mathbb{C}^M$ です。つまりこれは、未知のスカラー信号 x に関して、雑音を含んだ M 個の観測 y_1, y_2, \dots, y_M が得られている状況を意味します。直観的には複数の観測 y_1, y_2, \dots, y_M をうまく利用することで、 x についての観測が一つだけしか得られていない場合に比べて、より精度の高い x の推定値が得られそうですが、具体的には y_1, y_2, \dots, y_M をどのように利用すればよいでしょうか。

4.5.1 選 択 合 成

素朴な方法としては、 M 個の観測 y_1, y_2, \dots, y_M のうち最も品質のよいもの、言い換えると SNR の高いもの、だけを利用する方法があります。 m 番目の観測 y_m は

$$y_m = a_m x + v_m \quad (4.69)$$

と書け、右辺第1項が信号成分、第2項が雑音成分を表すので、そのSNRは

$$\gamma_m = \frac{E[|a_m|^2|x|^2]}{E[|v_m|^2]} = \frac{|a_m|^2\sigma_x^2}{\sigma_v^2} \quad (4.70)$$

となります。ただし、 $E[|x|^2] = \sigma_x^2$ 、 $E[|v_m|^2] = \sigma_v^2$ としています。これより

$$m_{\text{slct}} = \arg \max_{m=1,2,\dots,M} \gamma_m \quad (4.71)$$

$$= \arg \max_{m=1,2,\dots,M} |a_m|^2 \quad (4.72)$$

なるインデックスの観測信号 $y_{m_{\text{slct}}}$ が M 個の観測の中で最大の SNR をもち、これを利用することで観測が一つしか得られなかった場合に比べて精度の高い x の推定値を得ることができます。この方法は選択合成 (selection combining, SC) と呼ばれます[†]。

4.5.2 等利得合成

選択合成では SNR が最大でない観測信号はすべて破棄されてしまいますが、これらの信号を利用する合成方法も考えることができます。

単純に M 個の観測 y_1, y_2, \dots, y_M を足し合わせると、各観測の係数 a_m , $m = 1, 2, \dots, M$ の位相によっては希望信号成分が打ち消されてしまう可能性があります。そこで考えられた方法は、 M 個の観測を希望信号について同位相で足し合わせるというものです。具体的には

$$\begin{aligned} \hat{x}_{\text{eqi}} &= \mathbf{w}_{\text{eqi}}^H \mathbf{y} \\ &= \mathbf{w}_{\text{eqi}}^H a x + \mathbf{w}_{\text{eqi}}^H v \end{aligned} \quad (4.73)$$

によって合成します。ただし、合成の重みベクトル \mathbf{w}_{eqi} は

$$\mathbf{w}_{\text{eqi}} = [w_{\text{eqi},1} \ w_{\text{eqi},2} \ \dots \ w_{\text{eqi},M}]^T = \left[\frac{a_1}{|a_1|} \ \frac{a_2}{|a_2|} \ \dots \ \frac{a_M}{|a_M|} \right]^T \quad (4.74)$$

[†] 実際には信号を合成（足し合わせ）しませんが、慣例的にこのように呼んでいます。

で定義されます。ここで、 \mathbf{w}_{eqi} の各成分 $w_{\text{eqi},m}$, $m = 1, 2, \dots, M$ は各観測の係数 a_m , $m = 1, 2, \dots, M$ の位相で、 $|w_{\text{eqi},m}| = 1$ となり、 M 個の観測 y_1, y_2, \dots, y_M に対して同じ大きさの利得を与えて合成するので、**等利得合成** (equal-gain combining, **EGC**) と呼ばれます。

各観測の係数 a_m , $m = 1, 2, \dots, M$ と雑音 v_m , $m = 1, 2, \dots, M$ が無相関とすると、重みが乗算された $w_{\text{eqi},m}v_m$ も白色雑音となり、合成後の信号 \hat{x}_{eqi} 中の総雑音電力は $M\sigma_v^2$ となります。一方、希望信号成分はその係数が同相合成されるので、合成後の信号 \hat{x}_{eqi} の SNR は

$$\gamma_{\text{eqi}} = \frac{\left(\sum_{m=1}^M |a_m|\right)^2 \sigma_x^2}{M\sigma_v^2} \quad (4.75)$$

$$= \frac{1}{M} \left(\sum_{m=1}^M |a_m| \right)^2 \cdot \frac{\sigma_x^2}{\sigma_v^2} \quad (4.76)$$

$$= \frac{1}{M} \left(\sum_{m=1}^M \sqrt{\gamma_m} \right)^2 \quad (4.77)$$

となります。

等利得合成は、ハードウェアの制約などから信号の位相のみを変更できる場合などによく使用されます。

選択合成と等利得合成の合成後の SNR の大小は a_m の値、具体的には、「 $|a_m|^2$ の最大値」と「 $|a_m|$ の和の 2 乗を M で割ったもの」のどちらが大きいかに依存します。ある特定のインデックスの $|a_m|$ が他の係数の絶対値に比べて特に大きいような場合には選択合成が、最大の $|a_m|$ と同程度の絶対値の係数が多くあるような場合には等利得合成が、それぞれ有効であるといえます。

4.5.3 最 大 比 合 成

係数 a_m の値によらず、つねに選択合成や等利得合成と同等以上の SNR の信号が得られる方法として、**最大比合成** (maximal ratio combining, **MRC**) と呼ばれる合成法があり、信号合成の問題で最もよく利用されます。

これは M 個の観測 y_1, y_2, \dots, y_M を合成後の SNR が最大になるように線形に合成する方法で、 M 個の観測 y_1, y_2, \dots, y_M のそれぞれに対する重みを要素にもつベクトルを $\mathbf{w}_{\text{mrc}} \in \mathbb{C}^M$ とすると、合成後の信号はつぎで与えられます。

$$\begin{aligned}\hat{x}_{\text{mrc}} &= \mathbf{w}_{\text{mrc}}^H \mathbf{y} \\ &= \mathbf{w}_{\text{mrc}}^H \mathbf{a}x + \mathbf{w}_{\text{mrc}}^H \mathbf{v}\end{aligned}\quad (4.78)$$

ここで、2 行目右辺第 1 項が信号成分、第 2 項が雑音成分をそれぞれ表すため、線形合成後の信号の SNR は

$$\begin{aligned}\gamma_{\text{mrc}} &= \frac{E[|\mathbf{w}_{\text{mrc}}^H \mathbf{a}x|^2]}{E[|\mathbf{w}_{\text{mrc}}^H \mathbf{v}|^2]} \\ &= \frac{\sigma_x^2 \mathbf{w}_{\text{mrc}}^H \mathbf{a} \mathbf{a}^H \mathbf{w}_{\text{mrc}}}{\sigma_v^2 \mathbf{w}_{\text{mrc}}^H \mathbf{w}_{\text{mrc}}}\end{aligned}\quad (4.79)$$

となります。式 (4.79) より線形合成後の信号の SNR は行列 $\mathbf{a}\mathbf{a}^H$ のレイリー商（の定数倍）の形で表現されることがわかります[†]。このため、線形合成後の信号の SNR を最大にする \mathbf{w}_{mrc} は、2.4 節の性質 6 から $\mathbf{a}\mathbf{a}^H$ の最大固有値に対応する固有ベクトルによって与えられます。さらに、行列 $\mathbf{a}\mathbf{a}^H$ はすべての列ベクトルが \mathbf{a} の定数倍となっていることからランク 1 の行列であり、 \mathbf{a} は非零の固有値に対応する固有ベクトルであることがただちにわかります。

実際

$$\mathbf{a}\mathbf{a}^H \mathbf{a} = (|a_1|^2 + |a_2|^2 + \cdots + |a_M|^2) \mathbf{a} \quad (4.80)$$

より、 \mathbf{a} が $\mathbf{a}\mathbf{a}^H$ の固有ベクトルで、 $|a_1|^2 + |a_2|^2 + \cdots + |a_M|^2$ が対応する固有値であることが確認できます。したがって、最大比合成の重みベクトルとして

$$\mathbf{w}_{\text{mrc}}^H = \mathbf{a}^H \quad (4.81)$$

[†] 任意の正方行列 \mathbf{A} に対する固有方程式 (characteristic equation) $\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$ について考えてみます。両辺に左から \mathbf{w}^H を乗算すると $\mathbf{w}^H \mathbf{A}\mathbf{w} = \lambda \mathbf{w}^H \mathbf{w}$ となることから、行列 \mathbf{A} の任意の固有値は $\lambda = \frac{\mathbf{w}^H \mathbf{A}\mathbf{w}}{\mathbf{w}^H \mathbf{w}}$ なる形で表現されることがわかります。これは式 (2.59) に出てきたレイリー商です。

とすればよいことがわかります。

最大比合成後の信号の SNR は

$$\begin{aligned}\gamma_{\text{mrc}} &= \frac{\sigma_x^2 \mathbf{a}^H \mathbf{a} \mathbf{a}^H \mathbf{a}}{\sigma_v^2 \mathbf{a}^H \mathbf{a}} \\ &= \frac{\sigma_x^2 \mathbf{a}^H \mathbf{a}}{\sigma_v^2} \\ &= \frac{|a_1|^2 \sigma_x^2}{\sigma_v^2} + \frac{|a_2|^2 \sigma_x^2}{\sigma_v^2} + \cdots + \frac{|a_M|^2 \sigma_x^2}{\sigma_v^2}\end{aligned}\quad (4.82)$$

となります。これは最大比合成後の信号の SNR が個々の観測の SNR の和となることを意味し、最大比合成によって必ず SNR が改善する（正確には、劣化しない）ことがわかります。

これまで、式(4.68)の送信信号がスカラーのモデルについて考えてきましたが、式(4.1)のモデル

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v} \quad (4.83)$$

の場合にも同様のアイデアが適用できます。式(4.1)は

$$\mathbf{y} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_N x_N + \mathbf{v} \quad (4.84)$$

と書けるので、例えば \mathbf{y} から x_1 を推定する場合を考え、 x_2, x_3, \dots, x_N の成分を無視して最大比合成のアイデアを用いると

$$\hat{x}_{\text{mf},1} = \mathbf{a}_1^H \mathbf{y} \quad (4.85)$$

によって推定値が得られます。 \mathbf{x} の他の成分も同様に推定することにすると、 \mathbf{x} の推定値が

$$\hat{\mathbf{x}}_{\text{mf}} = \begin{bmatrix} \mathbf{a}_1^H \mathbf{y} \\ \mathbf{a}_2^H \mathbf{y} \\ \vdots \\ \mathbf{a}_N^H \mathbf{y} \end{bmatrix} = \mathbf{A}^H \mathbf{y} \quad (4.86)$$

によって得られます。これは \mathbf{x} の線形推定のための重み行列として \mathbf{A}^H を用いることを意味し、**整合フィルタ推定** (matched filter estimation, **MF 推定**) と呼ばれます。MF 推定は干渉成分の影響を考慮していないので、一般に ZF 推定や MMSE 推定に比べてその推定精度はよくありませんが、逆行列演算を必要としないので実装が容易です。

\mathbf{A} の各成分が平均 0 の**独立同分布** (independent and identically distributed, **i.i.d.**) に従うとし、 M と N の比率を保ったまま \mathbf{A} のサイズを大きくすると、任意の \mathbf{A} の実現値に対してそのグラム行列 (Gram matrix) $\mathbf{A}^H \mathbf{A}$ が対角行列に近づく（対角成分に比べて非対角成分が無視できるほどの大きさになる）ことが知られており、**大規模 MIMO** (massive MIMO) の文脈では**チャネル硬化** (channel hardening) と呼ばれます²⁸⁾。非常に大きなサイズの \mathbf{A} に対しては逆行列演算の必要な ZF 推定や MMSE 推定の適用が困難ですが、チャネル硬化を利用することで単純な MF 推定によって十分な精度の推定値が得られるため、MF 推定は大規模 MIMO などで応用上重要な信号推定法です。

4.6 最大事後確率推定と最尤推定

式 (4.1) の線形観測モデルにおいて、未知ベクトル \mathbf{x} の各成分が既知の有限のシンボル (symbol) 集合 (アルファベット (alphabet) といいます) \mathcal{S} から生成されるとなります。すなわち $\mathbf{x} \in \mathcal{S}^N$ です。この仮定は、未知ベクトルの各成分が任意の連続値をとる場合には成立しませんが、離散値のシンボルに情報を載せて信号を送信するデジタル通信の**信号検出** (signal detection) 問題や**機械学習** (machine learning) の**クラス分類** (classification) 問題などにあてはまります。

観測ベクトル \mathbf{y} が得られたとき、 \mathbf{x} の推定値として選んだ任意の $\hat{\mathbf{x}} \in \mathcal{S}^N$ が、真の \mathbf{x} と一致する確率は条件付き確率 $P(\hat{\mathbf{x}}|\mathbf{y})$ で与えられ、これは観測というイベントの後 (事後) の \mathbf{x} の確率という意味で**事後確率** (posterior probability) と呼ばれます。これに対して、観測ベクトル \mathbf{y} による条件付きでない確率 $P(\mathbf{x})$

は事前確率 (prior probability) と呼ばれます。

したがって、判定が正しい確率を最大にするようなルール、言い換えると、誤り確率 (error probability) が最小になるような判定ルールは、事後確率を最大にする $\hat{x} \in \mathcal{S}^N$ を選択することで実現されます。このルールを最適化問題の形で書くと

$$\hat{x}_{\text{map}} = \arg \max_{x \in \mathcal{S}^N} P(x|y) \quad (4.87)$$

となります。この推定法は最大事後確率推定 (maximum a posteriori probability estimation, MAP 推定) と呼ばれ、誤り確率を最小にするという意味で最適 (optimal) な検出方法です^{†1}。

ベイズの定理を用いると、事後確率は

$$P(x|y) = \frac{p(y|x)P(x)}{p(y)} \quad (4.88)$$

と書き直すことができます^{†2}。ここで、 x の各候補が同様に確からしく、事前分布 $P(x)$ が一様分布 (uniform distribution) であるとすると、事後確率を最大化する問題は

$$\hat{x}_{\text{ml}} = \arg \max_{x \in \mathcal{S}^N} p(y|x) \quad (4.89)$$

と書くことができます。これは尤度関数 (likelihood function) $p(y|x)$ を最大化する問題であることから最尤推定 (maximum likelihood estimation, ML 推定) と呼ばれます。

式 (4.1) の線形観測モデルにおいて v の各成分が独立同分布に従う複素ガウス雑音であるとすると、尤度関数は

$$p(y|x) = \frac{1}{\pi^M \det\{\mathbf{R}_v\}} \exp\left(-\frac{\|y - Ax\|_2^2}{\sigma_v^2}\right) \quad (4.90)$$

^{†1} 「最適」という言葉を使う際にはどのような規準（例えば、誤り率や平均 2 乗誤差など）で最適であるかをあわせて明記する必要があります。

^{†2} これは連続確率変数と離散確率変数についてのベイズの定理²⁹⁾であり、厳密には通常のベイズの定理（離散確率変数どうしや連続確率変数どうしを扱うベイズの定理）とは異なることに注意します。

と書けます。ここで、 $\mathbf{R}_v = E[\mathbf{v}\mathbf{v}^H] = \sigma_v^2 \mathbf{I}_M$ は加法性白色観測雑音ベクトルの相関行列です。これより、最尤推定の最適化問題は

$$\hat{\mathbf{x}}_{\text{ml}} = \arg \min_{\mathbf{x} \in \mathcal{S}^N} \|\mathbf{y} - \mathbf{Ax}\|_2^2 \quad (4.91)$$

と書き直すことができます。これは、観測雑音が加法性白色ガウス雑音 (additive white Gaussian noise, AWGN) の場合には、最尤推定は観測信号 \mathbf{y} からのユークリッド距離が最小となる \mathbf{Ax} を見つけ出す問題に帰着されることを意味します。

最尤推定の最適解を求めるためには、 \mathcal{S} の要素数を $|\mathcal{S}|$ とすると $|\mathcal{S}|^N$ 個ある \mathbf{x} のすべての候補についてユークリッド距離を評価する必要があり、指數オーダーの計算量となるため N が大きい場合には解を得ることが困難です。ただし、観測行列 \mathbf{A} が畳込み行列 (convolution matrix) と呼ばれる特別なテブリツ行列の場合には、ビタビアルゴリズム^{30), 31)} (Viterbi algorithm) という低演算量の解法アルゴリズムがあります。他にも、球面復号化³²⁾ (sphere decoding) などさまざまな計算量削減法が提案されていますが、一般に最尤推定に基づく手法はその良好な特性と引き換えに、線形推定と比較して要求演算量が大きいという欠点があります。

線形観測モデルに限らず、より一般的な問題設定での最大事後確率推定や最尤推定については、第 6 章と第 7 章で説明します。

4.7 通信応用の例

4.7.1 通信路等化

無線通信では、送信アンテナ (transmitting antenna, Tx. antenna) から送信された電波が受信アンテナ (receiving antenna, Rx. antenna) に到達するまでの経路 (path) は、直接到達するものや周辺のさまざまな物体に反射してから到達するものなど一般に複数存在します。このような通信路はマルチパス通信路 (multipath channel) と呼ばれます。

マルチパス通信路の各パス（経路）の長さは一般に異なるため、送信アンテナから受信アンテナに到達するまでの時間がパスごとに異なります。このため、送信アンテナから非常に時間幅の短いパルスを送信しても、受信側では時間的に広がった信号として観測されます。これを遅延広がり（delay spread）と呼びます。光通信においても波長分散（chromatic dispersion）や偏波モード分散（polarization mode dispersion）などの影響で遅延広がりが発生します。

遅延広がりが通信品質に及ぼす影響は、その通信路に送信される信号に依存します。例えば、遅延広がりに比べて十分長いシンボル周期（symbol period）をもつ送信シンボルを用いたデジタル通信では遅延広がりはほとんど通信品質に影響しません。一方、高速デジタル通信では一般に時間幅の短いシンボルが送信されますが、シンボル周期が遅延広がりと同等以下になると隣り合うシンボルどうしが干渉し、送信信号の波形とは似ても似つかないような受信信号が得られます。このような信号ひずみのことをシンボル間干渉（inter-symbol interference, ISI）と呼びます。

遅延広がりの影響が無視できず、ISI が生じるような通信路のことを周波数選択性通信路（frequency selective channel）と呼びます。これは、そのような通信路のインパルス応答（impulse response）をフーリエ変換して得られる周波数応答（frequency response）が、信号の帯域内で一定ではないためです（周波数によって応答が異なる = 周波数選択性）。逆に、シンボル周期に比べて遅延広がりが十分小さい通信路は周波数非選択性通信路（frequency non-selective channel）と呼ばれます。

上述のように深刻な ISI が生じるかどうかは送信される信号に依存するので、同じ物理的な通信路であっても、そこに送信する信号によって周波数選択性になったり周波数非選択性になったりします。図 4.1 に同じ遅延広がりをもつ通信路にシンボル周期の短い信号と長い信号を送信したときの ISI の影響を示します。いずれの場合も各パスに対応する遅延時間をもった信号の重ね合わせが受信信号となりますが、シンボル周期が短い場合は受信信号の波形が大きくひずんでいる一方、シンボル周期が長い場合にはシンボルの中心付近ではひずみ

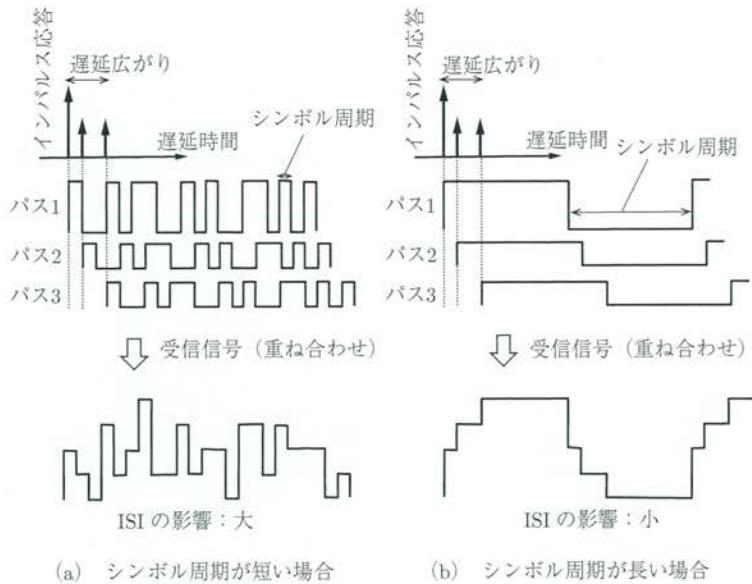


図 4.1 ISI とシンボル周期の関係

の影響がほとんどないことがわかります。

ISI によるひずみを受けた受信信号から、送信シンボルを推定するために ISI の影響を取り除く処理のことを通信路等化 (channel equalization) といいます。この処理を周波数領域で見ると、通信路の周波数選択性による応答の凹凸をフラットにする処理と解釈できるため、等化 (=各周波数における通信路の応答を等しくする) と呼ばれています。図 4.2 に通信路等化を用いたシステムのブロック図を示します。多くの場合、等化器は受信機に設置され、ISI による受信信号のひずみを補償することでその出力が送信信号に近くなるように制御

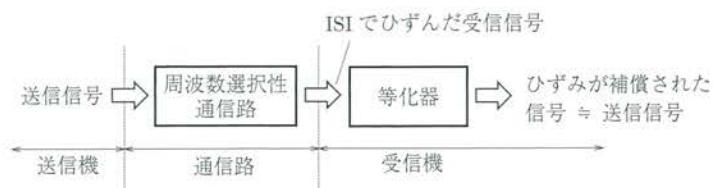


図 4.2 通信路等化

されます。通信路応答の情報が送信機にわかる場合は送信側で等化処理を行うことも可能です（**送信等化**（transmit equalization）と呼ばれます）。

周波数選択性通信路における送受信信号モデルは典型的に式(4.1)の線形観測モデルで表現されます。具体的には、周波数選択性通信路の離散時間インパルス応答を $\{h_0, h_1, \dots, h_L\}$ 、送信信号系列を $\{s_n\}$ 、観測雑音系列を $\{v_n\}$ とすると、受信信号系列 $\{r_n\}$ はインパルス応答と送信信号系列の畠込み演算に観測雑音を附加したもの、すなわち

$$r_n = \sum_{i=0}^L h_i s_{n-i} + v_n \quad (4.92)$$

によって与えられます。

さらに、 $s = [s_1 \ s_2 \ \dots \ s_N]$ を送信信号ベクトル、 $r = [r_1 \ r_2 \ \dots \ r_M]$ を受信信号ベクトル、 $v = [v_1 \ v_2 \ \dots \ v_M]$ を観測雑音ベクトルと定義すると、受信信号モデルは

$$r = Hs + v \quad (4.93)$$

と表されます。ただし、 H は通信路のインパルス応答で構成されるテプリツ構造をもった通信路行列で

$$H = \begin{bmatrix} h_0 & 0 & \dots & \dots & \dots & 0 \\ \vdots & h_0 & \ddots & & & \vdots \\ h_L & & \ddots & \ddots & & \vdots \\ 0 & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0 \\ 0 & \dots & 0 & h_L & \dots & h_0 \end{bmatrix} \in \mathbb{C}^{N \times N} \quad (4.94)$$

で与えられます。ここでは簡単のため、 $M = N$ とし $N > L$ であることを仮定しています。

式(4.93)の受信信号モデルは通常の線形観測モデルの形をしているので、これまでに学んださまざまな線形逆問題のための手法が適用できます。通常、通信路

等化のためには ZF 推定や MMSE 推定が用いられます。高速通信では重み行列を求める際の逆行列の演算量が実装上の問題となることがあります。このため、近年の多くの通信規格では周波数領域等化 (frequency domain equalization, FDE) と呼ばれる高速フーリエ変換 (fast Fourier transform, FFT) をうまく利用した低演算量の等化方法が広く採用されています。

周波数領域等化を用いた通信システムでは、複数のシンボルから構成される信号ブロック（ベクトル）を連続して送信し、受信側ではブロックごとに等化や復調の処理を行います。これまで考えてきた線形観測モデルの式でも未知の信号をベクトルとして扱ってきましたが、ここで考える周波数領域等化を用いたブロック伝送方式では複数のブロック間の干渉の影響を考慮する必要がある点が異なります。

ブロック長を Q 、 n 番目の送信信号ベクトルを $\bar{s}(n) = [\bar{s}_1(n) \ \bar{s}_2(n) \ \dots \ \bar{s}_Q(n)]^T$ とすると、 $\bar{s}(n)$ に対応する受信信号ベクトルは

$$\bar{r}(n) = [\bar{r}_1(n) \ \bar{r}_2(n) \ \dots \ \bar{r}_Q(n)]^T = \mathbf{H}_0 \begin{bmatrix} \bar{s}(n-1) \\ \bar{s}(n) \end{bmatrix} + \bar{v}(n) \quad (4.95)$$

と書くことができます。ただし、 $\bar{v}(n) = [\bar{v}_1(n) \ \bar{v}_2(n) \ \dots \ \bar{v}_Q(n)]^T$ は観測雑音ベクトルで、 \mathbf{H}_0 は

$$\mathbf{H}_0 = \begin{bmatrix} 0 & \dots & h_L & \dots & h_0 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \dots & 0 & h_L & \dots & h_0 \end{bmatrix} \in \mathbb{C}^{Q \times 2Q} \quad (4.96)$$

です。ここで、周波数選択性フェージングの影響により、 n 番目の送信信号ベクトル $\bar{s}(n)$ に対応する受信信号ベクトル $\bar{r}(n)$ に $n-1$ 番目の送信信号ベクトル $\bar{s}(n-1)$ の成分が含まれていることに注意します。また、 n 番目の受信信号ベクトルのとり方（タイミング）によっては、 $n-1$ 番目の送信信号ベクトルだけでなく、 $n+1$ 番目の送信信号ベクトル $\bar{s}(n+1)$ の成分も含まれることも

ありますが、ここでは、 $\bar{s}(n+1)$ の成分が入ってこないタイミングに同期されていることを仮定しています。

通信路行列 \mathbf{H}_0 を $Q \times Q$ の二つの部分行列

$$\mathbf{H}_2 = \begin{bmatrix} 0 & \dots & h_L & \dots & h_1 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & h_L \\ \vdots & & & & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \quad (4.97)$$

$$\mathbf{H}_1 = \begin{bmatrix} h_0 & & & \\ \vdots & h_0 & & \mathbf{0} \\ h_L & & \ddots & \\ & \ddots & \ddots & \\ \mathbf{0} & & h_L & \dots & h_0 \end{bmatrix} \quad (4.98)$$

に分解すると、受信信号ベクトル $\bar{r}(n)$ は

$$\bar{r}(n) = \mathbf{H}_2 \bar{s}(n-1) + \mathbf{H}_1 \bar{s}(n) + \bar{v}(n) \quad (4.99)$$

と書くことができます。

ここで右辺第1項は $n-1$ 番目の送信信号ベクトルからの成分であり、ブロック間干渉 (inter-block interference, IBI) を表しています。ブロック伝送ではガード区間 (guard interval) と呼ばれる新しい情報を伝送しない時間、言い換えると冗長な信号を送信する時間をブロックとブロックの間に挿入することで、ブロック間干渉を完全に除去できるようにします。

また、右辺第2項は希望信号、すなわち n 番目の送信信号ベクトルの成分から構成されますが、行列 \mathbf{H}_0 によってブロック内でのシンボル間干渉が生じています。周波数領域等化を用いたブロック伝送方式では、シンボル間干渉の影

響を低演算量で取り除くことができるようプロック間の冗長な信号が選ばれます。

プロック伝送においてプロック間干渉を除去する一つの方法として、少し乱暴ですが受信信号ベクトル $\bar{r}(n)$ の最初の K 個（ただし、 $K \geq L$ ）の成分を削除してしまうことが考えられます。この処理は、行列

$$\mathbf{R}_{\text{cp}} = \begin{bmatrix} \mathbf{0}_{(Q-K) \times K} & \mathbf{I}_{Q-K} \end{bmatrix} \in \mathbb{R}^{(Q-K) \times Q} \quad (4.100)$$

を $\bar{r}(n)$ の左から乗算することで実現されます。すなわち

$$\begin{aligned} r(n) &= \mathbf{R}_{\text{cp}} \bar{r}(n) \\ &= \mathbf{R}_{\text{cp}} \mathbf{H}_2 \bar{s}(n-1) + \mathbf{R}_{\text{cp}} \mathbf{H}_1 \bar{s}(n) + \mathbf{R}_{\text{cp}} \bar{v}(n) \\ &= \mathbf{R}_{\text{cp}} \mathbf{H}_1 \bar{s}(n) + v(n) \end{aligned} \quad (4.101)$$

です。ここで、 $v(n) = \mathbf{R}_{\text{cp}} \bar{v}(n)$ であり、 $K \geq L$ のときに $\mathbf{R}_{\text{cp}} \mathbf{H}_2$ が零行列になることを利用しています。

式 (4.101) の受信信号モデルではプロック間干渉の成分を完全に除去できますが、未知ベクトル $\bar{s}(n)$ の次元 Q に対して観測ベクトル $r(n)$ の次元が $Q - K$ しかないので $r(n)$ から $\bar{s}(n)$ を推定するのが難しいという問題があります。

そこで長さ Q の送信信号のプロック $\bar{s}(n)$ は、実は $Q - K (= M)$ 個の独立なシンボルからなるベクトル $s(n)$ に K 個の冗長な成分を付加したものと考えてみます。さらに、冗長成分の付加は $(M + K) \times M$ の行列 \mathbf{T}_{cp} の乗算によって行うとして $\bar{s}(n) = \mathbf{T}_{\text{cp}} s(n)$ とすると、式 (4.101) は

$$r(n) = \mathbf{R}_{\text{cp}} \mathbf{H}_1 \mathbf{T}_{\text{cp}} s(n) + v(n) \quad (4.102)$$

と書けます。行列 $\mathbf{R}_{\text{cp}} \mathbf{H}_1 \mathbf{T}_{\text{cp}}$ のサイズは $M \times M$ なので、 $\mathbf{R}_{\text{cp}} \mathbf{H}_1 \mathbf{T}_{\text{cp}}$ が正則であれば、 $r(n)$ から ZF 推定や MMSE 推定などで $s(n)$ を推定することができます。 $s(n)$ の推定に必要な計算量は行列 \mathbf{T}_{cp} の選び方に依存します。

周波数領域等化システムではサイクリックプレフィックス (cyclic prefix, CP) と呼ばれる冗長成分の付加方法が広く用いられています。これは $s(n)$ の最後の K 個の成分をそのままの順序で $s(n)$ の先頭に付加するもので、行列で表すと

$$\mathbf{T}_{\text{cp}} = \begin{bmatrix} \mathbf{0}_{K \times (M-K)} & \mathbf{I}_K \\ & \mathbf{I}_M \end{bmatrix} \quad (4.103)$$

となります。これを用いると行列 $\mathbf{R}_{\text{cp}} \mathbf{H}_1 \mathbf{T}_{\text{cp}}$ はつぎのような特殊な構造をもつ行列となります。

$$\mathbf{R}_{\text{cp}} \mathbf{H}_1 \mathbf{T}_{\text{cp}} = \begin{bmatrix} h_0 & 0 & \dots & 0 & h_L & \dots & h_1 \\ \vdots & h_0 & \ddots & & \ddots & \ddots & \vdots \\ h_L & & \ddots & \ddots & & \ddots & h_L \\ 0 & \ddots & & \ddots & \ddots & & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & 0 \\ 0 & \dots & 0 & h_L & \dots & h_0 \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{C}_{\text{cp}} \quad (4.104)$$

この行列の各列（行）は一つ前の列（行）全体を下（右）にシフトし、はみ出た成分を先頭にもってきたベクトルになっています。このような構造をもつ行列は巡回行列 (circulant matrix) と呼ばれます。巡回行列はその成分の値によらず離散フーリエ変換 (discrete Fourier transform, DFT) 行列

$$\mathbf{D} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{-j\frac{2\pi \times 1 \times 1}{M}} & \dots & e^{-j\frac{2\pi \times 1 \times (M-1)}{M}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j\frac{2\pi (M-1) \times 1}{M}} & \dots & e^{-j\frac{2\pi (M-1) \times (M-1)}{M}} \end{bmatrix} \quad (4.105)$$

によってユニタリ相似変換 (unitary similarity transform) が可能であるという非常に有用な性質をもっています^{33), 34)}。つまり

$$C_{\text{ep}} = D^H \Lambda D \quad (4.106)$$

と書くことができます。ただし Λ は

$$\begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{M-1} \end{bmatrix} = \sqrt{M} D \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \\ \mathbf{0}_{(M-L-1) \times 1} \end{bmatrix} \quad (4.107)$$

で定義される $\{\lambda_0, \lambda_1, \dots, \lambda_{M-1}\}$, すなわち通信路の周波数応答を対角成分にもつ対角行列です。また, D はユニタリ行列なので, $D^H = D^{-1}$ は逆離散フーリエ変換 (inverse discrete Fourier transform, IDFT) を表す行列であることに注意します。

以上より, 式 (4.102) は

$$\mathbf{r}(n) = D^H \Lambda D s(n) + \mathbf{v}(n) \quad (4.108)$$

と書くことができます。線形等化を仮定し, 等化器出力を

$$\hat{s}(n) = \mathbf{W}^H \mathbf{r}(n) \quad (4.109)$$

とすると, ZF 等化の場合には式 (4.10) より

$$\begin{aligned} \mathbf{W}^H &= (D^H \Lambda D)^{-1} \\ &= D^{-1} \Lambda^{-1} D^{-H} \\ &= D^H \Lambda^{-1} D \end{aligned} \quad (4.110)$$

となります。また, MMSE 等化の場合には, 式 (4.44) より

$$\begin{aligned} \mathbf{W}^H &= \sigma_s^2 \mathbf{H}^H (\sigma_s^2 \mathbf{H} \mathbf{H}^H + \sigma_v^2 \mathbf{I}_M)^{-1} \\ &= \sigma_s^2 \mathbf{D}^H \Lambda^H \mathbf{D} (\sigma_s^2 \mathbf{D}^H \Lambda \Lambda^H \mathbf{D} + \sigma_v^2 \mathbf{I}_M)^{-1} \end{aligned}$$

$$= \mathbf{D}^H \mathbf{\Lambda}^H \left(\mathbf{\Lambda} \mathbf{\Lambda}^H + \frac{\sigma_v^2}{\sigma_s^2} \mathbf{I}_M \right)^{-1} \mathbf{D} \quad (4.111)$$

となります。

ここで、等化器行列 \mathbf{W}^H は、ZF 等化と MMSE 等化のいずれの場合も、対角行列を IDFT の操作を表す \mathbf{D}^H と DFT の操作を表す \mathbf{D} で挟んだ形になっている点が重要です。受信信号ベクトルに対してまず \mathbf{D} を乗算することで周波数領域の信号に変換し、そこで通信路での信号ひずみを補償する対角行列を乗算した後、 \mathbf{D}^H の乗算によって再び信号を時間領域に戻すという処理になっていることから、これは周波数領域等化と呼ばれます。実際の通信システムでは、 \mathbf{D} や \mathbf{D}^H の乗算はそのまま行わず、FFT や逆 FFT のアルゴリズムを用いて効率的に計算されます。対角行列の逆行列は各対角成分の逆数を求めるだけで計算でき、また、対角行列の乗算はベクトルの各成分にそれぞれ対応する対角成分を乗算するだけで計算できるので、周波数領域等化はきわめて計算効率の高い等化方法です。

なお、ここまで説明してきた周波数領域等化の方式は、**CP を用いたシングルキャリヤブロック伝送方式** (single carrier block transmission with cyclic prefix, SC-CP) と呼ばれます。一方、同じ送受信モデルで送信信号ベクトルを $s(n)$ ではなく $\mathbf{D}^H s(n)$ とすると、よく知られた**OFDM** (orthogonal frequency division multiplexing) 方式が得られます。

式 (4.108)において送信信号ベクトルを $s(n)$ ではなく $\mathbf{D}^H s(n)$ であるとすると

$$\mathbf{r}(n) = \mathbf{D}^H \mathbf{\Lambda} s(n) + \mathbf{v}(n) \quad (4.112)$$

となりますが、この受信信号ベクトルに \mathbf{D} を乗算して離散周波数領域の信号に変換すると

$$\begin{aligned} \mathbf{R}(n) &= \mathbf{D} \mathbf{r}(n) \\ &= \mathbf{\Lambda} s(n) + \mathbf{V}(n) \end{aligned} \quad (4.113)$$

となります。ただし、 $\mathbf{V}(n) = \mathbf{D}\mathbf{v}(n)$ です。つまり、OFDM 方式の場合、周波数選択性フェージングの影響は送信信号ベクトル $\mathbf{s}(n)$ に対する対角行列 $\mathbf{\Lambda}$ の乗算として表され、SC-CP 方式同様、容易に等化を行うことができます。ここで、OFDM の場合は ZF 等化と MMSE 等化の重み行列が本質的に同一となり、その信号検出特性も同一になることに注意しましょう。

ここでは、ZF 推定と MMSE 推定に基づく線形等化の例を示しましたが、最尤推定に基づく**最尤系列推定** (maximum likelihood sequence estimation, MLSE) も等化基準としてよく用いられます。

4.7.2 通信路推定

無線通信における通信路等化や信号ひずみ補償、無線資源割り当てなどの処理は、多くの場合、通信路のインパルス応答や周波数応答に基づいて行われます。通信路のインパルス応答や周波数応答は一般に**通信路応答** (channel response) と呼ばれ、これを推定することを**通信路推定** (channel estimation) といいます。

通信路推定で考えるモデルは等化の場合と同様に

$$\mathbf{r} = \mathbf{H}\mathbf{p} + \mathbf{v} \quad (4.114)$$

で表されます。ただし、ここでは送信信号ベクトルとして既知の**パイロット信号** (pilot signal) ベクトル $\mathbf{p} = [p_1 \ p_2 \ \cdots \ p_M]^T$ を仮定していることに注意します。

さらに、等化の例と同様に CP を用いたブロック伝送方式を仮定して、通信路行列 \mathbf{H} が巡回行列であるとすると、巡回行列の性質から式 (4.114) は

$$\mathbf{r} = \mathbf{P}\mathbf{h} + \mathbf{v} \quad (4.115)$$

と書き直すことができます。ここで、 \mathbf{P} はパイロット信号で構成される巡回行列

$$\mathbf{P} = \begin{bmatrix} p_1 & p_M & \cdots & p_2 \\ p_2 & p_1 & \cdots & p_3 \\ \vdots & \vdots & \ddots & \vdots \\ p_M & p_{M-1} & \cdots & p_1 \end{bmatrix} \quad (4.116)$$

であり、 \mathbf{h} は \mathbf{H} の1列目のベクトル、すなわち、 $\mathbf{h} = [h_0 \ h_1 \ \cdots \ h_L \ 0 \ \cdots \ 0]^T$ です。

なお、CPを用いたブロック伝送方式を仮定しない場合にも、受信信号ベクトルのサイズを $\mathbf{r} \in \mathbb{C}^{M-L}$ 、通信路の畳込み行列を

$$\mathbf{H}_{\text{cnv}} = \begin{bmatrix} h_L & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & h_L & \cdots & h_0 \end{bmatrix} \in \mathbb{C}^{(M-L) \times M}$$

として

$$\mathbf{r} = \mathbf{H}_{\text{cnv}} \mathbf{p} + \mathbf{v} \quad (4.117)$$

と書くことができ、畳込み行列がもつテプリツ構造を利用してCPを用いたブロック伝送方式の場合と同様

$$\mathbf{r} = \mathbf{P}_{\text{cnv}} \mathbf{h}_{\text{cnv}} + \mathbf{v} \quad (4.118)$$

という形のモデルを得ることができます。ただし、この場合は、 $\mathbf{h}_{\text{cnv}} = [h_0 \ h_1 \ \cdots \ h_L]^T$ で

$$\mathbf{P}_{\text{cnv}} = \begin{bmatrix} p_{L+1} & p_L & \cdots & p_1 \\ p_{L+2} & p_{L+1} & \cdots & p_2 \\ \vdots & \vdots & \ddots & \vdots \\ p_M & p_{M-1} & \cdots & p_{M-L} \end{bmatrix} \in \mathbb{C}^{(M-L) \times (L+1)} \quad (4.119)$$

は正方行列ではないことに注意します。

P が巡回行列であるときに式 (4.115) から h を推定する問題は、式 (4.108) から s を推定する問題と本質的に同じです。このため、基本的には 4.7.1 項で説明した s の推定法と同様の推定法が適用可能です。ただし、等化の場合と異なり、 $E[hh^H]$ は通常事前にわからないため、通信路推定の問題では MMSE 推定ではなく P の逆行列を用いた ZF 推定（最小 2 乗推定）を用いることが多いです[†]。

インパルス応答 h の成分がスパースであることが事前にわかっている場合は次章で説明する圧縮センシングの手法がそのまま適用でき、これによって必要なパイロット信号数を減らす検討も行われています³⁶⁾。

4.7.3 MIMO 信号検出

MIMO 通信は複数のアンテナを備えた送受信機間で行う通信のことで、その受信信号モデルとして、そのまま式 (4.1) が利用できます。

送信アンテナ数を N 、受信アンテナ数を M とすると、 $s \in \mathbb{C}^N$ を送信信号ベクトル、 $r \in \mathbb{C}^M$ を受信信号ベクトル、 $H \in \mathbb{C}^{M \times N}$ を通信路行列、 $v \in \mathbb{C}^M$ を付加雑音ベクトルとして

$$r = Hs + v \quad (4.120)$$

と書けます。ここで、各アンテナ間の通信路はその利得が複素数のスカラーで表されるフラットフェージング通信路を仮定しています。通信路応答が畳込みで表される周波数選択性通信路の場合は、前述の周波数領域等化によるブロック伝送方式と組み合わせることで、式 (4.120) と同様のモデルが得られます。

この受信信号モデルにおいて、既知の受信信号ベクトル r と通信路行列 H の情報から送信信号ベクトル s を推定する問題は MIMO 信号検出と呼ばれます。式 (4.120) の MIMO 通信のモデルは、通信路等化や通信路推定の場合のそれ

[†] 何らかの方法で $E[hh^H]$ の情報を得ることで、MMSE 推定を利用して推定精度を向上させる方法も検討されています³⁵⁾。

と式の上では同一の形をしていますが、MIMO 通信の場合には通信路行列 \mathbf{H} に巡回性やテプリツ性などの確定的な構造を仮定することはほとんどなく、多くの場合、各成分が独立同分布に従うランダム行列として扱われます。このため、MIMO 通信の信号検出法はその多くが一般的で他の問題にも応用しやすく、特に文献37)などの MIMO 通信の教科書は通信の信号処理手法を効率的に学ぶのに適しているといえます。実際、MIMO 通信では本章で述べたほとんどの信号推定法がその検出法として検討されています。

最近では送受信アンテナ数が数十～数百といった大規模な MIMO 通信が検討されており、そこでは単にアンテナ数が多くなるだけでなく、まさに “more is different” というべき大規模システム特有の興味深い性質が見られます。大規模 MIMO に興味のある方は文献28)などを参照してください。

4.8 む す び

本章では、最初に線形観測モデルを導入し、その逆問題を解くための基本的な手法について説明しました。最後にいくつかの通信応用の例を示しましたが、実際、ほとんどの通信の信号処理の問題は線形観測モデルで記述されるので、本章で説明した各手法を理解すれば、世の中にある通信の信号処理手法のほとんどを理解できるといっても過言ではありません。この意味で、本章は本書のコアをなす章といえますので、本章で説明した手法の問題設定とその解法を、白紙に何も見ないで書いて説明できるようになるまで理解しましょう。その際、第2章で学んだ相関行列に関する性質と第3章で学んだウィルティンガー微分の計算ルールが非常に有用であることがよく理解できると思います。

5

圧縮センシング

Next SIP

本章では式(4.1)の線形観測モデルが**劣決定** (underdetermined, $M < N$) の場合を考えます。つまり、観測ベクトル \mathbf{y} の次元が未知ベクトル \mathbf{x} の次元よりも小さい場合です。このとき、式(4.8)を満足する \mathbf{W}_{zf} が存在しないので ZF 推定を行うことができません。一方、線形 MMSE 推定は $M < N$ の場合にも適用することが可能ですが、その推定精度は大きく劣化することが知られています。このような場合によく用いられるのが、**正則化** (regularization) と呼ばれるアプローチです。正則化では $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満足する \mathbf{x} のうち、その「大きさ」が最小のものを解として選択します。

5.1 最小ノルム解

正則化においてベクトルの大きさを測るノルムとして ℓ_2 ノルムを採用することにします。つまり、 $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満足する \mathbf{x} のうちで $\|\mathbf{x}\|_2$ が最小のものを選択するということです。この解は**最小ノルム解** (minimum norm solution) と呼ばれ、最適化問題

$$\hat{\mathbf{x}}_{\text{mn}} = \arg \min_{\mathbf{x} \in \mathbb{C}^N} \|\mathbf{x}\|_2^2, \quad \text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x} \quad (5.1)$$

を解くことで与えられます。この問題は ZF 推定のときと同様にラグランジュ未定乗数法を用いて解くことができます。未定乗数からなる M 次元のベクトルを $\boldsymbol{\phi}$ とし、ラグランジュ関数を

$$\mathcal{L}_{\text{mn}}(\mathbf{x}) = \|\mathbf{x}\|_2^2 + (\mathbf{A}\mathbf{x} - \mathbf{y})^\text{H} \boldsymbol{\phi} \quad (5.2)$$

とすると、 $\frac{\partial \mathcal{L}_{mn}(x)}{\partial x^H} = 0$ より

$$\hat{x}_{mn} = -A^H \phi \quad (5.3)$$

となります。これを制約式 $y = Ax$ に代入して、 A が行フルランクと仮定すると

$$\phi = -(AA^H)^{-1}y \quad (5.4)$$

となります。これより、最小ノルム解は

$$\hat{x}_{mn} = A^H(AA^H)^{-1}y \quad (5.5)$$

で与えられます。一見すると式(4.19)のZF推定(最小2乗推定)を用いたときの式とよく似ていますが、行列の並びが異なっていることに注意します。

最小ノルム解は観測雑音がある場合にも適用可能ですが、観測雑音の影響を陽には考えていませんでした。 $M < N$ の場合にも適用可能で、かつ観測雑音の影響を考慮した手法として**正則化最小2乗法**(regularized least squares method)があります。この推定値は、つぎの最適化問題を解くことで得られます。

$$\hat{x}_{rls} = \arg \min_x (\|Ax - y\|_2^2 + \lambda \|x\|_2^2) \quad (5.6)$$

$$= (\lambda I_N + A^H A)^{-1} A^H y \quad (5.7)$$

ここで、 $\lambda (\geq 0)$ は ℓ_2 正則化項 $\|x\|_2^2$ とデータ忠実(data fidelity)項 $\|Ax - y\|_2^2$ のバランスをとる正則化パラメータで、問題に応じてその値を適切に設定します。

最小ノルム解や正則化最小2乗法では**正則化項**(regularization term)として、未知ベクトルの ℓ_2 ノルムが用いられます。 ℓ_2 ノルムを用いると解が**閉形式**(closed form)で得られるメリットがありますが、正則化に ℓ_2 ノルム以外のノルムを用いない理由はありません。このことは後述する圧縮センシングと呼ばれる手法の伏線になっています。

5.2 スパース信号

圧縮センシング (compressed sensing) では、未知ベクトルがスパース (sparse) であること、すなわち非零要素の個数が見かけの次元に比べてずっと小さいこと、があらかじめわかっているときの未知ベクトル推定問題を考えます。このため、圧縮センシングの説明をする前に圧縮センシングの主役であるスパース信号（スパースベクトル）に関する基本事項を説明します。

簡単のため、本章ではこれ以降、すべての信号やベクトル、行列は実数値をとるものとします。もちろん圧縮センシングは複素数の信号に対しても拡張されています。興味のある方は文献38)などを参照してください。

ベクトルの非零要素の個数は ℓ_0 ノルムで表されるので、スパース信号 $\mathbf{x} \in \mathbb{R}^N$ は $\|\mathbf{x}\|_0 \ll N$ であるようなベクトルとして定義されます。ここで \ll は右辺が左辺よりはるかに大きいことを意味します。また、非零要素数を具体的に明示して $\|\mathbf{x}\|_0 \leq K$ なるベクトル、つまり非零要素数が高々 K 個のベクトルを定義して K -スパースなベクトルと呼ぶことがあります。このとき、 K -スパースなベクトル全体の集合を

$$\Sigma_K = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq K\} \quad (5.8)$$

で定義します。 $\mathbf{x}, \mathbf{z} \in \Sigma_K$ とすると、一般に $\mathbf{x} + \mathbf{z} \notin \Sigma_K$ なので集合 Σ_K は凸ではありませんが、 $\mathbf{x} + \mathbf{z} \in \Sigma_{2K}$ は成立することに注意します。

一方、多くの実際の問題では雑音やモデル化誤差 (modeling error) の影響などでスパース信号ではないがスパース信号でよく近似されるような信号を扱うことがあります。このような信号は近似的にスパースな信号 (approximately sparse signal)、あるいは圧縮可能信号 (compressible signal) などと呼ばれます。

ある信号 \mathbf{x} がスパース信号によってどの程度表現可能であるかを測る尺度として、つぎの量がよく用いられます。

$$\sigma_K(\mathbf{x})_p = \min_{\hat{\mathbf{x}} \in \Sigma_K} \|\mathbf{x} - \hat{\mathbf{x}}\|_p \quad (5.9)$$

これは \mathbf{x} を K -スパースな信号 $\hat{\mathbf{x}}$ で近似したときの誤差の大きさを ℓ_p ノルムで評価したときの最小値を意味し、 \mathbf{x} が実際に K -スパースな信号であれば p によらずその値が 0 になります。 $\sigma_K(\mathbf{x})_p$ は圧縮センシングの信号再構成における精度保証の尺度としてもよく用いられます。

任意の K -スパース信号 $\mathbf{x} \in \Sigma_K$ について、そのノルムの間にはつぎのような関係があり、圧縮センシングの文脈でしばしば利用されます。

$$\frac{\|\mathbf{x}\|_1}{\sqrt{K}} \leq \|\mathbf{x}\|_2 \leq \sqrt{K}\|\mathbf{x}\|_\infty \quad (5.10)$$

左側の不等式については、任意の K -スパース信号 \mathbf{x} について

$$\begin{aligned} \|\mathbf{x}\|_1 &= \mathbf{x}^\top \text{sgn}(\mathbf{x}) \\ &\leq \|\mathbf{x}\|_2 \cdot \|\text{sgn}(\mathbf{x})\|_2 \\ &\leq \sqrt{K}\|\mathbf{x}\|_2 \end{aligned} \quad (5.11)$$

となることから確認できます。ただし、 $\text{sgn}(\cdot)$ は符号関数 (sign function) で

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (5.12)$$

で定義され、引数がベクトルの場合は引数の成分ごとに上記の演算を施した結果を成分としてもつベクトルを出力するものとします。式 (5.11) の 1 行目から 2 行目への変形ではコーシー・シュワルツの不等式 (Cauchy-Schwartz inequality) を利用し、2 行目から 3 行目への変形では、任意の K -スパース信号 \mathbf{x} に対して $\text{sgn}(\mathbf{x}) \leq \sqrt{K}$ なることを利用します。

式 (5.10) の右側の不等式に現れる $\|\mathbf{x}\|_\infty$ は ℓ_∞ ノルムと呼ばれ

$$\|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \dots, |x_N|) \quad (5.13)$$

で定義されます。このため

$$\begin{aligned}
 \|\mathbf{x}\|_2^2 &= \sum_{n=1}^N |x_n|^2 \\
 &\leq K \{\max(|x_1|, |x_2|, \dots, |x_N|)\}^2 \\
 &= K \|\mathbf{x}\|_\infty^2
 \end{aligned} \tag{5.14}$$

となり、両辺の正の平方根をとることで右側の不等式が得られます。

5.3 圧縮センシングの考え方

未知ベクトル \mathbf{x} についての何らかの事前情報がある場合には、劣決定の線形観測 \mathbf{y} からでも観測された未知ベクトル \mathbf{x} を推定することができそうな気がします。実際、未知ベクトル \mathbf{x} がスパースであること、すなわち \mathbf{x} の ℓ_0 ノルムが見かけの次元 N に比べてずっと小さいこと、があらかじめわかっているときには、ある条件の下で $M < N$ の線形観測から実際に観測された未知ベクトル \mathbf{x} を見つけ出すことが可能です。この手法は圧縮センシング^{39)~41)} と呼ばれ、2000 年代中盤の誕生以降、さまざまな分野の問題に適用されています。

最小ノルム解による手法では ℓ_2 ノルムによる正則化を考えましたが、未知ベクトル \mathbf{x} がスパースであることが事前情報としてわかっている場合には、正則化のためのノルムとして ℓ_0 ノルムを用いるのが自然です。 ℓ_0 ノルムによる正則化を用いた手法は ℓ_0 再構成 (ℓ_0 reconstruction) と呼ばれ、最適化問題

$$\hat{\mathbf{x}}_{\ell_0} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_0, \quad s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{x} \tag{5.15}$$

を解くことで \mathbf{x} を推定します。

ℓ_0 再構成は素朴なアイデアに基づいていますが、非常に強力な手法です。実際、観測数 M が未知ベクトル \mathbf{x} の非零要素数よりも大きければ ℓ_0 再構成によってほとんどいつでも \mathbf{x} を再構成できることが知られています。しかし残念なことに、 ℓ_0 再構成は ℓ_0 ノルムの離散性 (discreteness) と非凸性 (non-convexity)

から一般に **NP 困難**⁴²⁾ (NP hard) であり、問題のサイズが大きくなると直接解くことが難しい問題になってしまいます。

このような場面でよくとられるのが、制約条件を緩めたり、目的関数を扱いやすい関数で近似したりする、**緩和** (relaxation) と呼ばれるアプローチです。 ℓ_0 再構成は**非凸最適化問題** (non-convex optimization problem) ですが、制約集合は元々凸集合なので、目的関数である ℓ_0 ノルムをこれに最も近い凸関数である ℓ_1 ノルムで置き換えることで緩和した**凸緩和問題** (convex relaxation problem)

$$\hat{x}_{\ell_1} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1, \quad s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (5.16)$$

を考えます。これは下に示すように**線形計画問題** (linear programming problem) に帰着され、現実的な計算量 (**多項式時間** (polynomial time)) で解くことができます。さらに重要なことに、 \mathbf{A} に関するある条件の下では、 $M < N$ であっても、この ℓ_1 緩和問題を解くことで実際に線形観測された真の解 \mathbf{x} が完全に再構成されることが理論的に示されています。

式(5.16)の ℓ_1 最適化問題において、二つのベクトル $\mathbf{u} = [u_1 \ u_2 \ \cdots \ u_N]^T, \mathbf{v} = [v_1 \ v_2 \ \cdots \ v_N]^T \in \mathbb{R}^N$ の成分をそれぞれ

$$u_n = \begin{cases} x_n, & x_n > 0 \\ 0, & x_n \leq 0 \end{cases} \quad (5.17)$$

$$v_n = \begin{cases} -x_n, & x_n < 0 \\ 0, & x_n \geq 0 \end{cases} \quad (5.18)$$

と定義すると、 $\mathbf{u} \geq 0, \mathbf{v} \geq 0, \mathbf{x} = \mathbf{u} - \mathbf{v}$ となります。ただし、ベクトルに対する ≥ 0 という記号は、ベクトルのすべての成分が非負であることを意味します。

さらに、 $\mathbf{z} = [\mathbf{u}^T, \mathbf{v}^T]^T$ と定義すると、コスト関数は

$$\|\mathbf{x}\|_1 = \mathbf{1}_N^T (\mathbf{u} + \mathbf{v}) = \mathbf{1}_{2N}^T \mathbf{z} \quad (5.19)$$

と書けます。また、制約式は

$$\mathbf{y} = \mathbf{Ax} = \mathbf{A}(\mathbf{u} - \mathbf{v}) = [\mathbf{A}, -\mathbf{A}]\mathbf{z} \quad (5.20)$$

となります。したがって、 ℓ_1 最適化問題は等価な問題として

$$\hat{\mathbf{z}}_{\text{lp}} = \arg \min_{\mathbf{z}} \mathbf{1}_{2N}^T \mathbf{z}, \quad s.t. \quad \mathbf{y} = [\mathbf{A}, -\mathbf{A}]\mathbf{z}, \quad \mathbf{z} \geq 0 \quad (5.21)$$

と書くことができ、これは線形計画問題の標準形となっています。

観測雑音の影響を考慮する場合には、等式制約 (equality constraint) を不等式制約 (inequality constraint) に置き換えた不等式 ℓ_2 制約付きの ℓ_1 最適化問題

$$\hat{\mathbf{x}}_{c\ell_1} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1, \quad s.t. \quad \|\mathbf{Ax} - \mathbf{y}\|_2^2 \leq \epsilon \quad (5.22)$$

を考えることができます。ここで、 $\epsilon > 0$ です。この不等式制約は観測雑音が有界 (bounded) の場合には自然な制約ですが、台が実数全体となるガウス雑音の場合にも式 (5.22) による再構成に理論的な精度保証を与えることができます。

また、式 (5.22) の最適化問題のラグランジュ関数を考えることで、ある $\mu > 0$ に対して式 (5.22) と同じ解を与える等価な制約なし最適化問題

$$\hat{\mathbf{x}}_{\ell_1-\ell_2} = \arg \min_{\mathbf{x}} \left(\mu \|\mathbf{x}\|_1 + \frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2 \right) \quad (5.23)$$

を導出できます。式 (5.23) はコスト関数に ℓ_1 ノルムと ℓ_2 ノルムの項が含まれていることから $\ell_1-\ell_2$ 最適化問題と呼ばれます。ただし、ある ϵ を用いた式 (5.22) と同じ解を与える式 (5.23) の μ の値を事前に知ることは、一般に難しいことに注意します。

式 (5.23) の最適化問題は、よく知られた **Lasso**⁴³⁾ (least absolute shrinkage and selection operator), すなわち

$$\hat{\mathbf{x}}_{\text{lasso}} = \arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{y}\|_2^2, \quad s.t. \quad \|\mathbf{x}\|_1 \leq t \quad (5.24)$$

のラグランジュ関数を考えることでも得られます。Lasso は、ある $t > 0$ に対して式 (5.22), (5.23) と同じ解を与える等価な再構成法です。文献によっては $\ell_1 - \ell_2$ 最適化のことを Lasso と呼んでいるものもありますが、Lasso のオリジナルの論文⁴³⁾ では式 (5.24) の不等式 ℓ_1 制約付き ℓ_2 正則化が提案されています。

5.4 再構成のアルゴリズム

圧縮センシングの信号再構成アルゴリズムには、大きく分けて

- 貪欲法 (greedy method) に基づく手法
- 近接分離 (proximal splitting) に基づく凸最適化手法
- 確率推論 (probabilistic inference) に基づく手法

などがあります。ここでは、圧縮センシングのアルゴリズムとして代表的な近接分離に基づく凸最適化手法のいくつかについて説明します。

近接分離に基づく凸最適化手法とは、近接写像 (proximity operator) と呼ばれる操作を利用した最適化アルゴリズム群のことをいいます⁴⁴⁾。ここで中心的な役割を果たす近接写像はつぎのように定義されます⁴⁵⁾。

定義 5.1 (近接写像) $f : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{\infty\}$ はプロパー (proper) な閉凸関数 (closed convex function), $\text{dom}(f)$ は関数 f の実効定義域 (effective domain) とする。このときパラメータ $\gamma > 0$ をもつ近接写像はつぎで与えられる。

$$\text{prox}_{\gamma f}(\mathbf{v}) = \arg \min_{\mathbf{x} \in \text{dom}(f)} \left\{ f(\mathbf{x}) + \frac{1}{2\gamma} \|\mathbf{x} - \mathbf{v}\|_2^2 \right\} \quad (5.25)$$

ここで、プロパーな閉凸関数とはエピグラフ (epigraph) と呼ばれる領域 $\text{epi}(f) = \{(\mathbf{x}, t) \in \mathbb{R}^N \times \mathbb{R} : f(\mathbf{x}) \leq t\}$ が空 (empty) でない閉凸集合 (closed convex set) である関数のことです。エピグラフは、直観的には、関数 f のグラフを描いたときにその上側にある点の集合のことです。また、実効定義域は

$\text{dom}(f) = \{\mathbf{x} \in \mathbb{R}^N : f(\mathbf{x}) < +\infty\}$ で定義されます。つまり、近接写像の定義では関数 f が ∞ の値をとることも許されますが、 $f(\mathbf{x})$ が有限の値となるような \mathbf{x} の集合を実効定義域と呼びます。

近接写像はそれ自体が最適化問題の形になっており、その意味を理解するのが少し難しいかもしれません。まず、 $\mathbf{v} \notin \text{dom}(f)$ の場合を考えます。このとき、 $\gamma \rightarrow 0$ であれば近接写像中のコスト関数は $\|\mathbf{x} - \mathbf{v}\|_2^2$ が支配的になるので、これを最小にする \mathbf{x} は \mathbf{v} の $\text{dom}(f)$ への直交射影となります。一方、 $\gamma \rightarrow \infty$ であれば近接写像中のコスト関数は $f(\mathbf{x})$ が支配的となり、近接写像は $f(\mathbf{x})$ の最小値を与える \mathbf{x} となります。このため、0 と ∞ の間の γ については、得られる近接写像もそれらの近接写像の間にある $\text{dom}(f)$ 内の点となります。 $\mathbf{v} \in \text{dom}(f)$ の場合には $\text{dom}(f)$ 内での写像となりますが、コスト関数の第2項で動ける範囲が制限されているので、元の \mathbf{v} からあまり離れない範囲で $f(\mathbf{x})$ の値をより小さくする $\text{dom}(f)$ 内の点が得られます。

ここで、圧縮センシングの問題で重要な二つの関数の近接写像の具体例を見てみましょう。一つ目は**指示関数** (indicator function) $\mathcal{I}_{\mathcal{C}}(\mathbf{x})$ で

$$\mathcal{I}_{\mathcal{C}}(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \mathcal{C} \\ \infty, & \mathbf{x} \notin \mathcal{C} \end{cases} \quad (5.26)$$

で定義されます。ここで、 $\mathcal{C} \subset \mathbb{R}^N$ は閉凸集合です。近接写像の定義に従って計算すると

$$\text{prox}_{\gamma \mathcal{I}_{\mathcal{C}}}(\mathbf{v}) = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \mathcal{I}_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2\gamma} \|\mathbf{x} - \mathbf{v}\|_2^2 \right\} \quad (5.27)$$

$$= \arg \min_{\mathbf{x} \in \mathcal{C}} \|\mathbf{x} - \mathbf{v}\|_2^2 \triangleq \Pi_{\mathcal{C}}(\mathbf{v}) \quad (5.28)$$

となり、閉凸集合 \mathcal{C} への直交射影となります。

つぎに ℓ_1 ノルム

$$f(\mathbf{x}) = \|\mathbf{x}\|_1 = \sum_{n=1}^N |x_n| \quad (5.29)$$

の近接写像を計算すると

$$\text{prox}_{\gamma f}(\mathbf{v}) = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \|\mathbf{x}\|_1 + \frac{1}{2\gamma} \|\mathbf{x} - \mathbf{v}\|_2^2 \right\} \quad (5.30)$$

$$= \arg \min_{\mathbf{x} \in \mathbb{R}^N} \sum_{n=1}^N \left\{ |x_n| + \frac{1}{2\gamma} (x_n - v_n)^2 \right\} \quad (5.31)$$

となり、要素ごとの最小化問題に帰着されます。要素ごとに考えるとコスト関数は2次関数ですので、絶対値記号について場合分けをすることで最適解が簡単に求まり、 $\text{prox}_{\gamma f}(\mathbf{v})$ の n 番目の成分は

$$\{\text{prox}_{\gamma f}(\mathbf{v})\}_n = \begin{cases} v_n - \gamma, & v_n \geq \gamma \\ 0, & -\gamma < v_n < \gamma \\ v_n + \gamma, & v_n \leq -\gamma \end{cases} \quad (5.32)$$

$$\triangleq S_\gamma(v_n) \quad (5.33)$$

となります。ここで、 v_n は \mathbf{v} の n 番目の要素で、関数 $S_\gamma(v_n)$ はソフトしきい値作用素 (soft thresholding operator) と呼ばれます。図 5.1 にソフトしきい値作用素のグラフを示します。

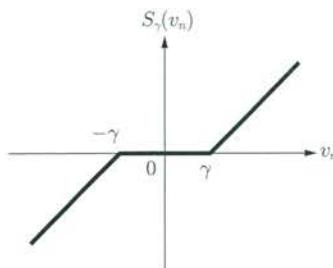


図 5.1 ソフトしきい値作用素

近接写像はどんな関数でも簡単に計算できるわけではありません。閉形式で得られる他の関数の近接写像については文献45)などを参照してください。

つぎに、近接写像を用いた代表的な最適化手法を三つ紹介します。一つ目は近接勾配法 (proximal gradient method) と呼ばれる手法で、つぎの最適化問題に対する解法アルゴリズムを与えます。

$$\min_{\mathbf{x} \in \mathbb{R}^N} \{\phi_1(\mathbf{x}) + \phi_2(\mathbf{x})\} \quad (5.34)$$

ここで、 $\phi_1(\cdot)$ はプロバーな閉凸関数で、 $\phi_2(\cdot)$ はリップシツ連続 (Lipschitz continuous) な勾配 $\nabla\phi_2$ をもつ微分可能 (differentiable) な凸関数です。この最適化問題を解くための近接勾配法は Algorithm 5.1 で与えられます⁴⁶⁾。

Algorithm 5.1 近接勾配法

初期値 \mathbf{x}_0 とパラメータ $\gamma > 0$ を与えて、以下を繰り返す。

$$\mathbf{x}_{k+1} = \text{prox}_{\gamma\phi_1}(\mathbf{x}_k - \gamma\nabla\phi_2(\mathbf{x}_k)) \quad (5.35)$$

ここで、 k は繰り返しのインデックス、 γ はステップサイズ、 $\nabla\phi_2(\mathbf{x}_k)$ は関数 $\phi_2(\cdot)$ の \mathbf{x}_k における勾配ベクトルを表します。勾配 $\nabla\phi_2$ のリップシツ定数を L とすると、 $\gamma \leq 1/L$ となるようにステップサイズを選択すればこのアルゴリズムによって得られるベクトルの列 $\{\mathbf{x}_k\}$ が式 (5.35) の最適化問題の解に収束することが知られています。また、その収束の速さは $O(1/k)$ であることも知られています。

近接勾配法では一つ前の推定値 \mathbf{x}_k を用いて更新値 \mathbf{x}_{k+1} を計算しますが、もう一つ前の推定値 \mathbf{x}_{k-1} を用いることで Algorithm 5.2 に示すような $O(1/k^2)$ で収束する高速なアルゴリズムを実現することができます⁴⁷⁾。

Algorithm 5.2 近接勾配法の高速版

初期値 $\mathbf{x}_0, \mathbf{r}_0, t_0$ とパラメータ $\gamma > 0$ を与えて、以下を繰り返す (ただし、通常 $t_0 = 1$)。

$$\mathbf{x}_{k+1} = \text{prox}_{\gamma\phi_1}(\mathbf{r}_k - \gamma\nabla\phi_2(\mathbf{r}_k)) \quad (5.36)$$

$$t_{k+1} = \frac{1}{2} \left(1 + \sqrt{4t_k^2 + 1} \right) \quad (5.37)$$

$$\mathbf{r}_{k+1} = \mathbf{x}_{k+1} + \left(1 + \frac{t_k - 1}{t_{k+1}} \right) (\mathbf{x}_{k+1} - \mathbf{x}_k) \quad (5.38)$$

近接勾配法を利用する際のポイントは、最適化問題に現れる二つの関数のうち、一つは微分可能で、もう一つは微分可能ではなくても近接写像が簡単に計算できる (prox 可能 (proximable) といいます) ことです。雑音がある場合の圧縮センシングで考えた $\ell_1-\ell_2$ 最適化

$$\hat{\mathbf{x}}_{\ell_1-\ell_2} = \arg \min_{\mathbf{x}} \left(\mu \|\mathbf{x}\|_1 + \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|_2^2 \right) \quad (5.39)$$

はその典型例で、実際

$$\phi_1(\mathbf{x}) = \lambda \|\mathbf{x}\|_1 \quad (5.40)$$

$$\phi_2(\mathbf{x}) = \frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2 \quad (5.41)$$

とおくと、 $\phi_1(\cdot)$ はプロパーな閉凸関数、 $\phi_2(\cdot)$ はリプシツ連続な勾配をもつ微分可能な凸関数となっています。 ℓ_1 ノルムの近接写像を用いることで $\ell_1-\ell_2$ 最適化のための近接勾配法である **ISTA** (iterative shrinkage and thresholding algorithm) とその高速版の **FISTA** (fast ISTA) の更新式はそれぞれつぎで与えられます。

ISTA の更新式：

$$\mathbf{x}_{k+1} = S_{\gamma\lambda}(\mathbf{x}_k - \gamma \mathbf{A}^\top (\mathbf{Ax}_k - \mathbf{y})) \quad (5.42)$$

FISTA の更新式：

$$\mathbf{x}_{k+1} = S_{\gamma\lambda}(\mathbf{z}_k - \gamma \mathbf{A}^\top (\mathbf{Az}_k - \mathbf{y})) \quad (5.43)$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \quad (5.44)$$

$$\mathbf{z}_{k+1} = \mathbf{x}_{k+1} + \frac{t_k - 1}{t_{k+1}} (\mathbf{x}_{k+1} - \mathbf{x}_k) \quad (5.45)$$

なお、オリジナルの文献47) では、一般の近接勾配法の高速版のアルゴリズムのことも FISTA と呼ばれることに注意します。

つぎに、Douglas-Rachford 分離と呼ばれる手法について説明します。近接勾配法のときと同様に

$$\min_{\mathbf{x} \in \mathbb{R}^N} \{\phi_1(\mathbf{x}) + \phi_2(\mathbf{x})\} \quad (5.46)$$

という最適化問題を考えますが、ここでは $\phi_1(\cdot)$ と $\phi_2(\cdot)$ のいずれもがプロパーな閉凸関数であればよく、いずれもが微分可能でなくてもよいことを仮定する点が異なっています。これを解くための Douglas-Rachford 分離アルゴリズムを Algorithm 5.3 に示します⁴⁸⁾。

Algorithm 5.3 Douglas-Rachford 分離アルゴリズム

初期値 \mathbf{r}_0 とパラメータ $\gamma > 0$, $\varepsilon \in (0, 1)$ を与えて、以下を繰り返す。

$$\mathbf{x}_k = \text{prox}_{\gamma\phi_1}(\mathbf{r}_k) \quad (5.47)$$

$$\eta_k \in [\varepsilon, 2 - \varepsilon] \quad (5.48)$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \eta_k (\text{prox}_{\gamma\phi_2}(2\mathbf{x}_k - \mathbf{r}_k) - \mathbf{x}_k) \quad (5.49)$$

指示関数を用いることで、等式制約付きの ℓ_1 最適化問題

$$\hat{\mathbf{x}}_{\ell_1} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_1, \quad s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (5.50)$$

を Douglas-Rachford 分離アルゴリズムによって解くことができます。具体的には、制約式を満足する \mathbf{x} の集合を $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^N | \mathbf{A}\mathbf{x} = \mathbf{y}\}$ と定義すると、指示関数 $\mathcal{I}_{\mathcal{C}}(\cdot)$ を用いて

$$\hat{\mathbf{x}}_{\ell_1} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} (\|\mathbf{x}\|_1 + \mathcal{I}_{\mathcal{C}}(\mathbf{x})) \quad (5.51)$$

と書くことができます。よって

$$\phi_1(\mathbf{x}) = \|\mathbf{x}\|_1 \quad (5.52)$$

$$\phi_2(\mathbf{x}) = \mathcal{I}_{\mathcal{C}}(\mathbf{x}) \quad (5.53)$$

とおけば、Douglas-Rachford 分離アルゴリズムが適用可能になります。

最後は**交互方向乗数法** (alternating direction method of multipliers, ADMM) と呼ばれる手法で

$$\min_{\mathbf{x} \in \mathbb{R}^L, \mathbf{z} \in \mathbb{R}^K} \{\phi_1(\mathbf{x}) + \phi_2(\mathbf{z})\}, \quad s.t. \quad \mathbf{z} = \mathbf{A}\mathbf{x} \quad (5.54)$$

なる最適化問題を考えます。ここで、 $\phi_1(\cdot)$, $\phi_2(\cdot)$ はプロバーな閉凸関数であり、 $\mathbf{A} \in \mathbb{R}^{K \times L}$ です。これに対する ADMM アルゴリズムは **Algorithm 5.4** で与えられます⁴⁸⁾。

Algorithm 5.4 ADMM アルゴリズム

初期値 \mathbf{z}_0 , \mathbf{v}_0 とパラメータ $\gamma > 0$ を与えて、以下を繰り返す。

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x} \in \mathbb{R}^L} \left\{ \phi_1(\mathbf{x}) + \frac{1}{2\gamma} \|\mathbf{A}\mathbf{x} - \mathbf{z}_k + \mathbf{v}_k\|^2 \right\} \quad (5.55)$$

$$\mathbf{z}_{k+1} = \text{prox}_{\gamma\phi_2}(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{v}_k) \quad (5.56)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{A}\mathbf{x}_{k+1} - \mathbf{z}_{k+1} \quad (5.57)$$

ADMM は幅広い分野のさまざまな問題に利用されていますが、圧縮センシングの文脈ではつぎのような最適化問題の解法アルゴリズムとして利用できます。

$$\min_{\mathbf{x} \in \mathbb{R}^N} \left(\frac{1}{2} \|\mathbf{x} - \mathbf{1}\|_1 + \frac{1}{2} \|\mathbf{x} + \mathbf{1}\|_1 + \lambda \|\mathbf{y} - \mathbf{Hx}\|_2^2 \right) \quad (5.58)$$

この問題は、未知ベクトル \mathbf{x} の各成分が 1 または -1 であることがあらかじめわかっているときに、未知ベクトルの離散性を利用してその線形観測から未知ベクトルを推定するためのもので **SOAV 最適化** (sum-of-absolute-value optimization) と呼ばれます⁴⁹⁾。

未知ベクトル \mathbf{x} 自体はスパースでなくとも、 $\mathbf{x} - \mathbf{1}$ や $\mathbf{x} + \mathbf{1}$ がスパース性をもつことをうまく利用することで、劣決定の線形観測から未知の離散値ベクトルがある条件の下で完全再構成できることが知られています。

ADMM を利用してこの問題を解くために、まずつぎのように変形します。

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^N} \left(\frac{1}{2} \|\mathbf{z}_1 - \mathbf{1}\|_1 + \frac{1}{2} \|\mathbf{z}_2 + \mathbf{1}\|_1 + \lambda \|\mathbf{y} - \mathbf{Hx}\|_2^2 \right), \\ & \text{s.t. } \mathbf{x} = \mathbf{z}_1, \quad \mathbf{x} = \mathbf{z}_2 \end{aligned} \quad (5.59)$$

さらに、 $\mathbf{z} = [\mathbf{z}_1^\top \ \mathbf{z}_2^\top]^\top$ 、 $\mathbf{A} = [\mathbf{I}_N \ \mathbf{I}_N]^\top$ として

$$\phi_1(\mathbf{x}) = \lambda \|\mathbf{y} - \mathbf{Hx}\|_2^2 \quad (5.60)$$

$$\phi_2(\mathbf{z}) = \frac{1}{2} \|\mathbf{z}_1 - \mathbf{1}\|_1 + \frac{1}{2} \|\mathbf{z}_2 + \mathbf{1}\|_1 \quad (5.61)$$

と定義すると

$$\min_{\mathbf{x} \in \mathbb{R}^N, \mathbf{z} \in \mathbb{R}^{2N}} \{\phi_1(\mathbf{x}) + \phi_2(\mathbf{z})\}, \quad \text{s.t. } \mathbf{z} = \mathbf{Ax} \quad (5.62)$$

となります。これは上記の ADMM の最適化問題の形になっています。

デジタル通信システムでは一般に送信信号が離散性を有するため、ここで説明した離散値ベクトル再構成のアイデアを利用した信号検出法などが検討されています^{38), 50)~52)}。

5.5 再構成の条件

圧縮センシングによって実際に観測されたスパース信号が再構成できるための条件について考えます。ここでは、簡単のために観測雑音がないものとします。

未知の K -スパースベクトル $\mathbf{x} \in \Sigma_K$ に対する線形観測を $\mathbf{y} = \mathbf{A}\mathbf{x}$ とすると、 $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満足する K -スパースなベクトル \mathbf{x} が高々一つ存在するなら観測ベクトル \mathbf{y} から \mathbf{x} を完全に再構成することができます。これについて、つぎのような完全再構成 (perfect reconstruction) の条件があります^{53), 54)}。

定理 5.1 (スパーク (spark) による完全再構成の条件) 任意の $\mathbf{y} \in \mathbb{R}^M$ に対して $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満足する $\mathbf{x} \in \Sigma_K$ が高々一つ存在する $\Leftrightarrow \text{spark}(\mathbf{A}) > 2K$

ここで、 $\text{spark}(\mathbf{A})$ は行列 \mathbf{A} のスパークと呼ばれる量で、 \mathbf{A} からとってきた列ベクトルの組が 1 次従属となる最小列数として定義されます[†]。この命題が成立することを確認してみましょう。まず $\text{spark}(\mathbf{A}) > 2K$ が完全再構成の**必要条件** (necessary condition) であること (\Rightarrow) を**背理法** (proof by contradiction) を用いて示します。

任意の $\mathbf{y} \in \mathbb{R}^M$ に対して $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満足する $\mathbf{x} \in \Sigma_K$ が高々一つ存在するとします。ここで、 Σ_K はすべての K -スパースなベクトルの集合です。このとき、 $\text{spark}(\mathbf{A}) \leq 2K$ であると仮定すると、これは \mathbf{A} に 1 次従属な $2K$ 個の列ベクトルの組が存在することを意味します。したがって、これらの $2K$ 個の列ベクトルに対応する成分に台をもつ $2K$ -スパースなベクトル $\mathbf{h} \in \Sigma_{2K}$ で、かつ $\mathbf{h} \in \mathcal{N}(\mathbf{A})$ 、すなわち $\mathbf{A}\mathbf{h} = \mathbf{0}$ となるものが存在します。ここで、二つの異なる K -スパースなベクトル $\mathbf{x}, \mathbf{x}' \in \Sigma_K$ を用いて $\mathbf{h} = \mathbf{x} - \mathbf{x}'$ と書けるので、

[†] これは **k-ランク** (Kruskal rank) とも呼ばれ、**テンソル分解** (tensor decomposition) の一意性の条件などにも現れます⁵⁵⁾。

$A(x - x') = 0$, すなわち $Ax = Ax'$ となります。これは $y = Ax$ を満足する $x \in \Sigma_K$ が高々一つ存在することに矛盾します。したがって, $\text{spark}(A) > 2K$ が示せました。

つぎに十分条件 (sufficient condition) であること (\Leftarrow) を示します。

$\text{spark}(A) > 2K$ が成立するとします。このとき, ある y について, $y = Ax = Ax'$ となる $x, x' \in \Sigma_K$ が存在するとして, $h = x - x'$ とおくと $Ah = 0$ が成立します。ここで, $\text{spark}(A) > 2K$ より A の任意の $2K$ 個の列ベクトルは 1 次独立なので, $h = 0$ となります。これは, $x = x'$ を意味し, よって, $\text{spark}(A) > 2K$ が成立するとき $y = Ax$ を満足する $x \in \Sigma_K$ が高々一つ存在することが示せました。

スパークによる完全再構成の条件では, 未知ベクトルが真にスパースであることが前提となっていましたが, つぎの制限等長性 (restricted isometry property, RIP) と呼ばれる性質を用いると圧縮可能信号も含めた再構成の精度保証を与えることができます⁵⁶⁾。

定義 5.2 (制限等長性 (RIP)) 行列 A が次数 K の RIP を満足することは, 任意の $x \in \Sigma_K$ に対して, つぎを満たす定数 $\delta_K \in (0, 1)$ が存在することである。

$$(1 - \delta_K)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_K)\|x\|_2^2 \quad (5.63)$$

RIP の意味するところは, K -スパースなベクトルの長さが行列 A の乗算によって大きく変化しないということです。例えば $\delta_K = 0$ の場合には, 不等式の両辺にある係数 $1 - \delta_K$ と $1 + \delta_K$ がどちらも 1 になるので, $\|x\|$ と $\|Ax\|$ が等しいという条件になります。

$x \in \Sigma_K$ の台 $\Lambda = \text{supp}(x)$ が既知とすると, x の台だけで構成されるベクトルを $x_\Lambda \in \mathbb{R}^K$, x の台に対応する A の列だけで構成される行列を $A_\Lambda \in \mathbb{R}^{M \times K}$ として

$$y = Ax = A_\Lambda x_\Lambda \quad (5.64)$$

と書けます。このとき、 A_Λ が良条件 (well-conditioned, 最大特異値と最小特異値の比で定義される条件数 (condition number) が大きくない) ならば、最小 2 乗法

$$x_\Lambda = (A_\Lambda^\top A_\Lambda)^{-1} A_\Lambda^\top y \quad (5.65)$$

により x_Λ を求めることができます[†]。

この推定法は、実際にはわからない（神様だけが知っている） $x \in \Sigma_K$ の台がわかると仮定しているという意味でオラクル推定量 (oracle estimator) と呼ばれます。この文脈では、RIP は $|\Lambda| = K$ なる任意の A の部分行列 A_Λ に對して良条件性を要請するものと解釈することができます。

さて、RIP を用いた再構成の精度保証を示す前に、圧縮可能信号の再構成の精度保証がどのような形で与えられるか見てみましょう。任意の再構成アルゴリズムによる推定値を $\hat{x}(y, A)$ とし、式 (5.9) を用いると、任意のベクトル x について、つぎの不等式が成立することを示すことで再構成の精度保証が与えられます。

$$\|\hat{x}(y, A) - x\|_2 \leq C \frac{\sigma_K(x)_1}{\sqrt{K}} \quad (5.66)$$

これは、左辺は推定誤差ベクトルの大きさを ℓ_2 ノルムで計ったもので、それが右辺の量、すなわち、 x を K -スパースなベクトルで近似したときの誤差の大きさを ℓ_1 ノルムで計ったときの最小値の定数 (C/\sqrt{K}) 倍で上から押さえられる、というものです。

左辺はよいとして、右辺については少し違和感を覚えられた方も多いと思います。左辺の誤差は ℓ_2 ノルムで計っているので、右辺の近似誤差も ℓ_2 ノルムで計るのが自然な気がします。つまり

[†] 観測に雑音が含まれず、かつ理想的な計算機が利用できるなら、悪条件 (ill-conditioned) でも（条件数が大きくても）問題ありませんが、実際には観測に雑音が含まれたり、数值計算の精度に限界があるので、悪条件では最小 2 乗法による推定が困難です。

$$\|\hat{x}(y, A) - x\|_2 \leq C\sigma_K(x)_2 \quad (5.67)$$

としてはどうでしょうか。実は、この精度保証を満足するために必要な観測数 M が未知ベクトルの次元 N に近くなってしまうという問題があります。そこで、任意の K -スパースなベクトルについて $\|x\|_1/\sqrt{K} \leq \|x\|_2$ なる関係があることを思い出して $\sigma_K(x)_2$ を $\sigma_K(x)_1/\sqrt{K}$ で置き換えることで、上記の精度保証の式 (5.66) が得られます。そのような都合で勝手に置き換えてよいのかということが気になりますが、 x が真に K -スパースな場合は、 $\sigma_K(x)_2$ も $\sigma_K(x)_1$ もどちらも 0 になるので、式 (5.66) が成立するなら任意の K -スパースなベクトルの完全再構成が保証でき、さらに圧縮可能信号などの非スパース信号の再構成についてもロバスト性が保証されます。

RIP による再構成の条件はつぎのように与えられます⁵⁶⁾。

定理 5.2 (RIP による ℓ_1 再構成の条件) 観測行列 A が定数 $\delta_{2K} < \sqrt{2} - 1$ 、次数 $2K$ の RIP を満足するとし、任意の $x_0 \in \mathbb{R}^N$ について、その線形観測 $y = Ax_0$ が得られたとする。このとき

$$\hat{x}_{\ell_1} = \arg \min_x \|x\|_1, \quad s.t. \quad y = Ax \quad (5.68)$$

の解 \hat{x}_{ℓ_1} はつぎを満足する。

$$\|\hat{x}_{\ell_1} - x_0\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} \quad (5.69)$$

ただし

$$C_0 = 2 \frac{1 - (1 - \sqrt{2})\delta_{2K}}{1 - (1 + \sqrt{2})\delta_{2K}} \quad (5.70)$$

である。

これは、 ℓ_1 再構成によって K -スパースなベクトルの完全再構成が可能であることを示しています。ただし、RIP は再構成の十分条件なので、RIP を満足

しない観測行列を用いたときにも ℓ_1 再構成によって完全再構成が達成される場合があることに注意します。

RIP を用いることで、 ℓ_0 再構成による完全再構成の十分条件もつぎのように与えられます⁵⁶⁾。

定理 5.3 (RIP による ℓ_0 再構成の条件) 観測行列 A が次数 $2K$ の RIP を満足するとし、任意の K -スパースなベクトル $x_0 \in \Sigma_K$ について、その線形観測 $y = Ax_0$ が得られたとする。このとき

$$\hat{x}_{\ell_0} = \arg \min_x \|x\|_0, \quad s.t. \quad y = Ax \quad (5.71)$$

は唯一の解 \hat{x}_{ℓ_0} をもち、 $\hat{x}_{\ell_0} = x_0$ となる。

ある行列が与えられたときに、個別の行列が RIP を満足するかどうかを評価することは一般に困難です。そこで、ランダム行列の集合に対して RIP が調べられています。例えば以下に示すような $M \times N$ のランダム行列は、 $M > \mathcal{O}(K \log N / K)$ を満足するなら高い確率で ℓ_1 再構成の十分条件となる RIP を満足することが知られています。

- 各成分が i.i.d. ガウス分布に従うランダム行列
- 各成分が i.i.d. ベルヌーイ分布 (Bernoulli distribution) に従うランダム行列
- 離散コサイン変換 (discrete cosine transform, DCT) 行列 (複素信号の場合には、DFT 行列) からランダムに行を抽出した部分行列

これは観測行列としてランダム行列の実現値を用いるだけです。観測対象のスパースベクトルに合わせて適応的に観測行列を選択する必要がないことを意味しています。このことは、応用上大変有用な性質です。

RIP を用いると再構成の十分条件が与えられましたが、つぎで定義される零空間特性 (null space property, NSP) を用いると必要十分条件が与えられます⁵⁴⁾。

定義 5.3 (零空間特性 (NSP)) Λ をインデックス $\{1, 2, \dots, N\}$ の部分集合, Λ^c を Λ の補集合, \mathbf{h}_Λ を Λ^c に含まれるインデックスの成分は 0 でそれ以外の成分は \mathbf{h} と同一のベクトルとする。行列 \mathbf{A} が次数 K の NSP を満足するとは、任意の $\mathbf{h} \in \mathcal{N}(\mathbf{A})$, $|\Lambda| \leq K$ である任意の Λ に対して、つぎを満たす定数 $C > 0$ が存在することである。

$$\|\mathbf{h}_\Lambda\|_2 \leq C \frac{\|\mathbf{h}_{\Lambda^c}\|_1}{\sqrt{K}} \quad (5.72)$$

NSP は直観的には、 \mathbf{A} の零空間のベクトルが少ないインデックス集合上に集中しないということを意味します。NSP を用いた再構成の必要条件は、例えばつぎで与えられます⁵⁷⁾。

定理 5.4 (NSP による再構成の必要条件) 観測行列 \mathbf{A} と任意の再構成アルゴリズムによる推定値 $\hat{\mathbf{x}}(\mathbf{y}, \mathbf{A})$ のペア $(\mathbf{A}, \hat{\mathbf{x}})$ を考える。 $(\mathbf{A}, \hat{\mathbf{x}})$ が任意の $\mathbf{x}_0 \in \mathbb{R}^N$ とその線形観測 $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ について

$$\|\hat{\mathbf{x}}(\mathbf{y}, \mathbf{A}) - \mathbf{x}_0\|_2 \leq C \frac{\sigma_K(\mathbf{x}_0)_1}{\sqrt{K}} \quad (5.73)$$

を満足するならば、 \mathbf{A} は次数 $2K$ の NSP を満足する。

この命題が成立することは簡単に確認できます。観測行列 \mathbf{A} の零空間の任意のベクトル $\mathbf{h} \in \mathcal{N}(\mathbf{A})$ を考え、その大きい $2K$ 個の要素のインデックス集合を Λ とします。さらに、集合 Λ を K 個ずつの集合 Λ_0 と Λ_1 に分けて

$$\mathbf{x}_0 = \mathbf{h}_{\Lambda_1} + \mathbf{h}_{\Lambda^c} \quad (5.74)$$

$$\mathbf{x}'_0 = -\mathbf{h}_{\Lambda_0} \quad (5.75)$$

とします。ここで、 $\mathbf{x}'_0 \in \Sigma_K$ なので、精度保証の不等式が成立するなら $\mathbf{x}'_0 = \hat{\mathbf{x}}(\mathbf{A}\mathbf{x}'_0, \mathbf{A})$ となります。さらに、 $\mathbf{h} = \mathbf{x}_0 - \mathbf{x}'_0$ および $\mathbf{h} \in \mathcal{N}(\mathbf{A})$ なので

$Ah = A(x_0 - x'_0) = \mathbf{0}$ となり、 $Ax_0 = Ax'_0$ 。したがって $x'_0 = \hat{x}(Ax_0, A)$ となります。以上より

$$\|h_\Lambda\|_2 \leq \|h\|_2 \quad (5.76)$$

$$= \|x_0 - x'_0\|_2 \quad (5.77)$$

$$= \|x_0 - \hat{x}(Ax_0, A)\|_2 \quad (5.78)$$

$$\leq C \frac{\sigma_K(x_0)_1}{\sqrt{K}} \quad (5.79)$$

$$= \sqrt{2}C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{2K}} \quad (5.80)$$

が得られます。ただし、3行目から4行目への変形では精度保証の不等式を利用し、4行目から5行目の変形では x_0 の値の大きい K 個の成分はすべて h_{Λ_1} に含まれていることを利用しています。式(5.80)は A が次数 $2K$ の NSP を満足することを意味しています。

なお、ここでは NSP が任意の再構成アルゴリズムの精度保証の必要条件であることを示しましたが、NSP は ℓ_1 再構成の精度保証の十分条件であることも知られています。

スパークや RIP、NSP は与えられた観測行列がそれらを満足するかどうかを評価すること自体が一般に困難です。このため、個別の観測行列に対してそれが満足するかどうかを容易に確認できるような再構成の条件があると大変便利です。つぎで定義される相互コヒーレンス (mutual coherence) はそのような目的でよく利用されます⁵³⁾。

定義 5.4 (相互コヒーレンス) 行列 $A \in \mathbb{R}^{M \times N}$ の相互コヒーレンス $\mu(A)$ は、 A の異なる列ベクトル間の正規化された内積の絶対値の最大値である。

$$\mu(A) = \max_{1 \leq i, j \leq N, i \neq j} \frac{|a_i^\top a_j|}{\|a_i\|_2 \cdot \|a_j\|_2} \quad (5.81)$$

ただし、 a_i は A の i 番目の列ベクトルである。

相互コヒーレンスとスパークの間には、任意の行列 A について

$$\text{spark}(A) \geq 1 + \frac{1}{\mu(A)} \quad (5.82)$$

なる関係が成り立ちます。これはつぎのように確認できます。

まず、スパークは列ベクトルの長さに依存しないので、簡単のために A は単位ノルムの列ベクトルをもつものとします。 Λ を $|\Lambda| = p$ なる列のインデックス $\{1, 2, \dots, N\}$ の部分集合とし、 Λ に含まれるインデックスに対応する A の列ベクトルだけで構成される行列 A_Λ のグラム行列 $G \in \mathbb{R}^{p \times p}$ を

$$G = A_\Lambda^\top A_\Lambda \quad (5.83)$$

で定義すると、 G の (i, j) 成分 $g_{i,j}$ についてつぎが成り立ちます。

$$g_{i,i} = 1, \quad 1 \leq i \leq p \quad (5.84)$$

$$|g_{i,j}| \leq \mu(A), \quad 1 \leq i, j \leq p, \quad i \neq j \quad (5.85)$$

もし、 G が正定値行列ならば、 A_Λ の列ベクトルは 1 次独立となります。そして、これがすべての $|\Lambda| = p$ なる列インデックス部分集合について成り立つなら、定義より $\text{spark}(A) > p$ を意味します。ここで、 G は実対称行列なのでその固有値は実数であり、**ゲルシュゴリンの定理** (Gershgorin circle theorem, 付録 A.3 参照) より $\sum_{j \neq i} |g_{i,j}| < |g_{i,i}|$ ならば G のすべての固有値が正となり G は正定値となります。式 (5.85) を用いると

$$\sum_{j \neq i} |g_{i,j}| \leq (p-1)\mu(A) \quad (5.86)$$

*5章が2行だけ
こぼれてしまうのを避けるため
これを1行としました。*

$$p < 1 + \frac{1}{\mu(A)}$$

を満足すれば \mathbf{G} が正定値となります。この条件は p が最大で $1 + \lfloor 1/\mu(\mathbf{A}) \rfloor$ にまでなり得ることを意味し、よって

$$\text{spark}(\mathbf{A}) > 1 + \left\lfloor \frac{1}{\mu(\mathbf{A})} \right\rfloor \quad (5.88)$$

すなわち、式 (5.82) が成立します。

式 (5.82) の関係を用いてスパークによる完全再構成の条件の $\text{spark}(\mathbf{A})$ を $1 + 1/\mu(\mathbf{A})$ で置き換えることで、相互コヒーレンスによる再構成の条件を与えることができます⁵³⁾。

定理 5.5 (相互コヒーレンスによる再構成の十分条件) 観測行列 \mathbf{A} の相互コヒーレンス $\mu(\mathbf{A})$ について

$$K < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{A})} \right) \quad (5.89)$$

が成立するならば、任意の \mathbf{y} に対して、 $\mathbf{y} = \mathbf{Ax}$ を満足する $\mathbf{x} \in \Sigma_K$ が高々一つ存在する。

ここで、この条件はスパークとは異なり、必要十分条件ではなく十分条件になっていることに注意します。その代わり相互コヒーレンスは比較的低演算量で評価できる点が大きなメリットとなっています。

5.6 むすび

本章では、観測ベクトルの次元が未知ベクトルの次元よりも小さい場合の線形観測モデルにおける逆問題について考えました。特に、未知ベクトルがスペースであることが仮定できる場合の手法は圧縮センシングやスペースモデリング (sparse modeling) と呼ばれ、いまではさまざまな分野で必要不可欠な手法となっています。通信分野の信号処理でも当然のように利用されていますので、よく理解しておきましょう。より詳しく学びたい方には文献53), 54), 58) がお勧めです。

6

部分空間法

Next SIP

本書で説明するほとんどの手法では式(4.1)の線形観測モデルにおけるベクトル x の推定問題を考えますが、本章では観測行列 A の成分あるいは A に含まれるパラメータを y の相関行列から推定することを考えます。具体的には、すでに学んだ「相関行列がエルミート行列になること」および「エルミート行列の異なる固有値に対応する固有ベクトルが直交すること」を利用した部分空間法(subspace method)と呼ばれる手法について説明します。部分空間法は様々な信号に対して利用可能ですが、通信の問題では空間中に配置された複数のセンサ素子(アンテナなど)で受信された信号に対する処理、すなわちアレー信号処理[†](array signal processing)の文脈で用いられることが多いため、アレー信号処理の基礎についての説明から始めます。

6.1 アレー信号処理の基礎

本節では、アレー信号処理における空間領域の信号モデルについて説明し、そのモデルを時間領域の信号処理における時系列のモデルと比較することで、空間領域のモデルにおける狭帯域信号(narrowband signal)の到来方向(direction of arrival, DOA)が時間領域のモデルにおける複素正弦波(complex sinusoidal)の角周波数(angular frequency)に対応する¹²⁾ことを示します。

時間領域の信号として、角周波数および複素振幅(complex amplitude)が

[†] 「アレイ信号処理」と表記されることもあります。

それぞれ $\omega_0, \omega_1, \dots, \omega_{L-1}$ と s_0, s_1, \dots, s_{L-1} である L 個の複素正弦波の重ね合わせからなる離散時間信号

$$r(n) = \sum_{l=0}^{L-1} s_l e^{j\omega_l n} + v(n), \quad n = 0, 1, \dots, N-1 \quad (6.1)$$

を考えます。ただし、 n は時刻であり、 $v(n)$ は白色付加雑音、 $N (> L)$ はサンプル数を表します。ここで、 $\omega_0, \omega_1, \dots, \omega_{L-1}$ を確定的な未知のパラメータ、 s_0, s_1, \dots, s_{L-1} を確率変数としたときに、 $r(n)$ から $\omega_0, \omega_1, \dots, \omega_{L-1}$ を推定する問題は、時間領域における古典的かつ典型的な複素正弦波の周波数推定問題 (frequency estimation problem) で、多くの応用に現れます。

つぎに、図 6.1 に示すような、 N 個のセンサ (sensor) を直線上に配置したセンサアレー (linear sensor array) で観測された信号を用いた空間信号処理について考えてみます。各センサは無指向性 (omnidirectional) とし、0 番目のセンサの位置 d_0 を原点として、 n 番目のセンサの位置を d_n とします。また、 $L (< N)$ 個の波源 (source) からの周波数 f_l の複素正弦波が、それぞれ異なる方向 θ_l , $-\pi/2 \leq \theta_l \leq \pi/2$, $l = 0, 1, \dots, L-1$ からセンサアレーに到来しているとします (図 6.1 では到来角 θ から到来する一波のみを図示しています)。

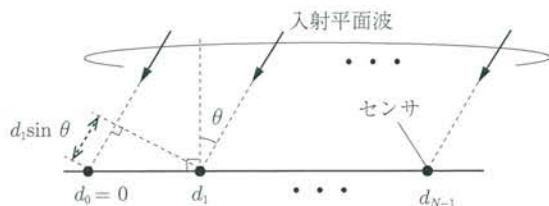


図 6.1 直線センサアレー

波源から放出された波の等位相面 (equiphasic surface) は一般に球面になりますが、波源とセンサアレーが十分離れていることを仮定し、センサアレー付近での到来波の等位相面を平面で近似します (平面波近似 (plane wave approximation) といいます)。このとき、0 番目のセンサと n 番目のセンサにおける l 番目の波源からの到来波の受信信号の相違は、観測雑音を除いて、行路差

(path difference) $d_n \sin \theta_l$ に対応する伝搬時間差 $(d_n \sin \theta_l)/c$ に起因する位相差 (phase difference)

$$\phi_l(n) = 2\pi \frac{d_n \sin \theta_l}{\lambda} \quad (6.2)$$

となります。ここで、 c は波の伝搬速度、 λ は到來波の波長であり、 $\lambda = c/f_0$ なる関係があります。

これより、 n 番目のセンサでの時刻 k における等価低域系 (equivalent low-pass system) の受信信号は

$$r(n, k) = \sum_{l=0}^{L-1} s_l e^{j\phi_l(n)} + v(n, k) \quad (6.3)$$

と表されます。ただし、 s_l は l 番目の波源からの到來波の 0 番目のセンサでの複素振幅であり、 $v(n, k)$ は時空間的に白色な付加雑音です。

特別な場合として、センサが直線上に等間隔 d で配置されている場合を考えてみます。

$$\phi_l = 2\pi \frac{d \sin \theta_l}{\lambda} \quad (6.4)$$

と定義すると、0 番目のセンサと n 番目のセンサにおける l 番目の波源からの到來波の位相差は $\phi_l(n) = \phi_l n$ となることから、式 (6.3) は

$$r(n, k) = \sum_{l=0}^{L-1} s_l e^{j\phi_l n} + v(n, k) \quad (6.5)$$

と書くことができます。

式 (6.5) の受信信号モデルは n と k の二つの変数を引数にもつ関数になっていますが、 k をある値に固定し n のみを引数にもつ関数とみなすと (k を固定したときのすべてのセンサでの受信信号からなるベクトルを、時刻 k のスナップショット (snapshot) と呼びます)、式 (6.1) の時間領域の信号モデルと式 (6.5) の空間領域の信号モデルは等価なモデルになっていることが確認できます。つまり、式 (6.1) において時系列 $r(n)$ 、 $n = 0, 1, \dots, N - 1$ から周波数 ω_l を推

定する問題と、式(6.5)において空間軸上の系列 $r(n, k)$, $n = 0, 1, \dots, N - 1$ から ϕ_l を推定する問題は本質的に同じ問題となります。

さらに、 $-\pi/2 \leq \theta_l \leq \pi/2$ より ϕ_l は区間 $(-2\pi d/\lambda, 2\pi d/\lambda)$ 内に存在することから、センサ間隔 (sensor spacing) d を $\lambda/2$ 以下に設定することで ϕ_l と θ_l が一対一対応となり、 ϕ_l から θ_l をユニークに決定することができます。これより、時系列に含まれる正弦波の周波数を推定する問題と、センサアレーの受信信号から到来波の到来角を推定する問題は本質的に同じであるともいえます。なお、ここでの ϕ_l から θ_l をユニークに決定するためのセンサ間隔 d に対する条件は、時間領域の信号処理におけるサンプリング定理 (sampling theorem)，すなわちサンプリング周期を T としたとき、サンプリング前の連続時間信号が角周波数 $(-\pi/T, \pi/T)$ に帯域制限 (band limited) されていれば元の信号がサンプルから完全に復元できること⁵⁹⁾ に対応しています。

ここまで、センサアレーの受信信号モデルとして、各波源から複素正弦波が送出される場合について考えてきました。この場合、各波源からの送信信号が定数倍の関係をもつことから、コヒーレント (coherent) 波[†]が到来するモデルと呼ばれます。実は、6.4節以降で説明する MUSIC 法を含む多くの到来方向推定の手法では、到来波がコヒーレントでないこと（インコヒーレント (incoherent) といいます）が仮定されます。無線通信システムの波源からの信号は純粋な正弦波ではなく搬送波周波数 (carrier frequency) f_0 をもつ変調信号 (modulated signal) であり、また、到来方向ごとに異なる遅延時間 (delay time) で到来するため、多くの通信の問題では到来波がインコヒーレントであることが仮定できます。

このとき、式(6.3)の受信信号モデルでは、 l 番目の波源からの到来波の複素振幅が定数ではなく時刻 k に依存するため、 s_l を $s_l(k)$ といった表記に変更する必要があります。さらに、厳密には、0 番目のセンサと n 番目のセンサにおける

[†] 二つあるいはそれ以上の波が、同じ周波数で位相差が一定であるとき、言い換えると、ある波を複素数の定数倍することで他の波が表されるとき、コヒーレントであるといいます。

l 番目の波源からの到来波の受信信号の相違は、搬送波における位相差だけでなく、0番目のセンサと n 番目のセンサの間の伝搬時間差 $d_n \sin \theta_l / c$ の間に $s_l(k)$ が変化すること、すなわち、 n 番目のセンサにおいては $s_l(k + d_n \sin \theta_l / c)$ となることも考慮する必要があります。この場合の受信信号モデルは複雑になってしまいますが、到来波が狭帯域でセンサアレー上を伝搬する間にその複素包絡線 (complex envelop) が大きく変化しないこと (つまり、 $s_l(k + d_n \sin \theta_l / c) \approx s_l(k)$ が成り立つこと) を仮定すると、受信信号モデルは式 (6.3) と同様に

$$r(n, k) = \sum_{l=0}^{L-1} s_l(k) e^{j\phi_l(n)} + v(n, k) \quad (6.6)$$

となります。

無線通信では、多くの場合、搬送波周波数が高く信号の狭帯域性が仮定できるため、通信のためのアレー信号処理ではこの信号モデルがよく利用されます。ただし、音波の信号は一般に電波を利用した無線通信の信号に比べて比帯域 (fractional bandwidth) が大きく、マイクロホンアレーを用いた信号処理などではこのモデルが適切でないことが多いので注意しましょう。

6.2 信号部分空間と雑音部分空間

式 (6.1) や式 (6.6) の信号モデルを一般に

$$r(n) = \sum_{l=0}^{L-1} s_l a_l(n) + v(n) \quad (6.7)$$

と書き、このモデルを用いて**信号部分空間** (signal subspace) と**雑音部分空間**⁶⁰⁾ (noise subspace) について説明します。

$$\mathbf{r} = [r(0) \ r(1) \ \cdots \ r(N-1)]^T \quad (6.8)$$

$$\mathbf{a}_l = [a_l(0) \ a_l(1) \ \cdots \ a_l(N-1)]^T \quad (6.9)$$

$$\mathbf{v} = [v(0) \ v(1) \ \cdots \ v(N-1)]^T \quad (6.10)$$

とおくと

$$\mathbf{r} = [\mathbf{a}_0 \ \mathbf{a}_1 \ \cdots \ \mathbf{a}_{L-1}] \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{L-1} \end{bmatrix} + \mathbf{v} \quad (6.11)$$

$$= \mathbf{A}\mathbf{s} + \mathbf{v} \quad (6.12)$$

と書けます。ただし

$$\mathbf{A} = [\mathbf{a}_0 \ \mathbf{a}_1 \ \cdots \ \mathbf{a}_{L-1}] \quad (6.13)$$

$$\mathbf{s} = [s_0 \ s_1 \ \cdots \ s_{L-1}]^T \quad (6.14)$$

です。

ここで、雑音ベクトル \mathbf{v} は平均 $E[\mathbf{v}] = \mathbf{0}$ 、共分散行列

$$E[\mathbf{v}\mathbf{v}^H] = \sigma_v^2 \mathbf{I}_N \quad (6.15)$$

とします。また、信号ベクトル \mathbf{s} は平均 $E[\mathbf{s}] = \mathbf{0}$ 、共分散行列

$$\mathbf{R}_s = E[\mathbf{s}\mathbf{s}^H] \quad (6.16)$$

で、 $\text{rank } \mathbf{R}_s = L$ とし、雑音と信号が無相関、すなわち、 $E[s_l v(n)^*] = 0$ と仮定します。このとき、観測ベクトル \mathbf{r} の相関行列は

$$\mathbf{R}_r = E[(\mathbf{A}\mathbf{s} + \mathbf{v})(\mathbf{A}\mathbf{s} + \mathbf{v})^H] \quad (6.17)$$

$$= \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_v^2 \mathbf{I}_N \quad (6.18)$$

となります。

すでに第2章で学んだように、相関行列の固有値はすべて非負の実数であり、 N 個のたがいに直交する固有ベクトルをもちます。つまり、 \mathbf{R}_r の固有値を大きさの順に並べて $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1} (\geq 0)$ とし、対応する正規化固有ベクトルを $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_{N-1}$ とすると

$$\mathbf{q}_i^H \mathbf{q}_j = \delta_{ij}, \quad i, j = 0, 1, \dots, N-1 \quad (6.19)$$

が成り立ちます。

同様に、 $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ の固有値と固有ベクトルについて考えると、 $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ は $\mathbf{A}\mathbf{s}$ の相関行列なので、その固有値は非負の実数であり、 N 個のたがいに直交する固有ベクトルをもちます。これらの固有値を大きさの順に並べて $\gamma_0 \geq \gamma_1 \geq \dots \geq \gamma_{N-1} (\geq 0)$ とします。

このとき

$$\mathbf{R}_r \mathbf{q}_n = \lambda_n \mathbf{q}_n \quad (6.20)$$

となります。一方

$$\mathbf{R}_r \mathbf{q}_n = (\mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_v^2 \mathbf{I}_N) \mathbf{q}_n \quad (6.21)$$

$$= \mathbf{A}\mathbf{R}_s\mathbf{A}^H \mathbf{q}_n + \sigma_v^2 \mathbf{q}_n \quad (6.22)$$

となることから

$$\lambda_n \mathbf{q}_n = \mathbf{A}\mathbf{R}_s\mathbf{A}^H \mathbf{q}_n + \sigma_v^2 \mathbf{q}_n \quad (6.23)$$

すなわち

$$\mathbf{A}\mathbf{R}_s\mathbf{A}^H \mathbf{q}_n = (\lambda_n - \sigma_v^2) \mathbf{q}_n \quad (6.24)$$

が成立するので、 \mathbf{R}_r の固有ベクトル \mathbf{q}_n は $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ の固有ベクトルでもあることがわかります。

式 (6.24) の固有方程式より \mathbf{q}_n に対応する $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ の固有値は $\lambda_n - \sigma_v^2$ ですが、 σ_v^2 が n に依存しない定数であることから、 $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ の固有値を大きさの順に並べると $\lambda_0 - \sigma_v^2 \geq \lambda_1 - \sigma_v^2 \geq \dots \geq \lambda_{N-1} - \sigma_v^2$ となり、元の定義の $\gamma_0 \geq \gamma_1 \geq \dots \geq \gamma_{N-1}$ と見比べると

$$\lambda_n = \gamma_n + \sigma_v^2 \quad (6.25)$$

が成り立つことがわかります。

ここで、 $N \times L$, $N > L$ の行列 \mathbf{A} が列フルランク、すなわち $\text{rank } \mathbf{A} = L$ であることを仮定します。このとき、 $\text{rank } \mathbf{R}_s = L$ が仮定されていることから、 $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ の小さいほうから $N - L$ 個の固有値はすべて 0 となります。したがって、式 (6.25) はさらに

$$\lambda_n = \begin{cases} \gamma_n + \sigma_v^2, & n = 0, 1, \dots, L-1 \\ \sigma_v^2, & n = L, L+1, \dots, N-1 \end{cases} \quad (6.26)$$

となります。

$\text{rank } \mathbf{A} = L$ より、 $\text{rank } \mathbf{A}^H = L$ であり、 \mathbf{A}^H のカーネル空間[†] (kernel space) $\mathcal{N}(\mathbf{A}^H)$ の次元は $N - L$ となります。ここで、 $\mathbf{z} \in \mathcal{N}(\mathbf{A}^H)$ に対しては $\mathbf{R}_r \mathbf{z} = \sigma^2 \mathbf{z}$ となるので、 \mathbf{z} は \mathbf{R}_r の固有値 σ^2 に対応する固有ベクトルとなります。このカーネル空間の次元は $N - L$ なので式 (6.26) のように重複度 (multiplicity) は $N - L$ であり、 $N - L$ 個の固有ベクトル $\mathbf{q}_L, \mathbf{q}_{L+1}, \dots, \mathbf{q}_{N-1}$ は $\mathcal{N}(\mathbf{A}^H)$ の正規直交基底となり、 $\mathbf{A}^H \mathbf{q}_n = \mathbf{0}$ 、すなわち

$$\mathbf{q}_n^H \mathbf{A} = \mathbf{0}, \quad n = L, L+1, \dots, N-1 \quad (6.27)$$

が成り立ちます。

ここで、 $\mathbf{Q}_S = [\mathbf{q}_0 \ \mathbf{q}_1 \ \cdots \ \mathbf{q}_{L-1}]$, $\mathbf{Q}_N = [\mathbf{q}_L \ \mathbf{q}_{L+1} \ \cdots \ \mathbf{q}_{N-1}]$ と定義すると、 \mathbf{Q}_S の列ベクトルの張る空間 $\mathcal{R}(\mathbf{Q}_S)$ は信号部分空間、 \mathbf{Q}_N の列ベクトルの張る空間 $\mathcal{R}(\mathbf{Q}_N)$ は雑音部分空間と呼ばれます⁶⁰⁾。これは、式 (6.27) より $\mathcal{R}(\mathbf{Q}_N) = \mathcal{N}(\mathbf{A}^H)$ であり、また $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_{N-1}$ は正規直交基底であることから $\mathcal{R}(\mathbf{Q}_S) = \mathcal{R}(\mathbf{Q}_N)^\perp$ となります ($^\perp$ は直交補空間 (orthogonal complement) を表します)、一般に $\mathcal{R}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^H)^\perp$ であることから

$$\mathcal{R}(\mathbf{Q}_S) = \mathcal{R}(\mathbf{A}) \quad (6.28)$$

[†] $\mathbf{A}^H \mathbf{z} = \mathbf{0}$ となるようなベクトル \mathbf{z} の集合をカーネル空間といいます。

および

$$\mathcal{R}(Q_N) = \mathcal{R}(A)^\perp \quad (6.29)$$

が成立するからです。なお、観測雑音はすべての部分空間に存在し、 $\mathcal{R}(Q_S)$ には信号成分だけでなく雑音成分も含まれることから、文献によっては信号部分空間のことを**信号プラス雑音部分空間** (signal-plus-noise subspace) と呼ぶこともあります。

6.3 主成分分析とマイナー成分分析

式(6.28)や式(6.29)の関係は重要かつ有用で、通信だけでなく多くの分野の信号処理に利用されています。特に重要なのは、現実の問題では式(6.12)の線形観測モデルの観測行列 A が未知であることが多い一方、多数の観測ベクトル $r_k, k = 0, 1, \dots, K-1$ を集めて、観測ベクトルのサンプル相関行列 (sample correlation matrix)

$$\hat{R}_r = \frac{1}{K} \sum_{k=0}^{K-1} r_k r_k^H \quad (6.30)$$

を計算し、その固有ベクトルの組を求めることで、信号部分空間や雑音部分空間を推定できるという点です。

一般に、**トレーニング信号** (training signal) などの既知の信号に対する観測が得られないと行列 A の成分を直接的に推定することはできませんが、観測ベクトルの相関行列の推定値さえ得られれば、行列 A の列ベクトルの張る空間 $\mathcal{R}(A)$ (やその直交補空間) ならば $\mathcal{R}(Q_S)$ から推定することができるというわけです。

推定された信号部分空間の利用法として、観測ベクトルからの**雑音除去** (denoising) があります。観測された r には観測雑音 v が付加されていますが、信号成分は行列 A の列ベクトルの張る空間 $\mathcal{R}(A)$ 内にしか存在せず、その補空間には雑音成分しか存在しません。このため、 r から最も近い $\mathcal{R}(A)$ の要素を

雑音成分の抑圧された観測信号 \hat{r} とすることが自然な考え方であり、これは r の $\mathcal{R}(A)$ への直交射影（orthogonal projection）にほかなりません。

$\mathcal{R}(A)$ の代わりに $\mathcal{R}(Q_S)$ を用いると

$$\begin{aligned}\hat{r} &= Q_S(Q_S^H Q_S)^{-1} Q_S^H r \\ &= Q_S Q_S^H r\end{aligned}\quad (6.31)$$

となります。ここで、 Q_S の各列はたがいに直交することから $Q_S^H Q_S = I_L$ となることを利用しています。さらに、成分を用いて展開すると式 (6.31) は

$$\hat{r} = [q_0 \ q_1 \ \cdots \ q_{L-1}] \begin{bmatrix} q_0^H \\ q_1^H \\ \vdots \\ q_{L-1}^H \end{bmatrix} r \quad (6.32)$$

$$= [q_0 \ q_1 \ \cdots \ q_{L-1}] \begin{bmatrix} q_0^H r \\ q_1^H r \\ \vdots \\ q_{L-1}^H r \end{bmatrix} \quad (6.33)$$

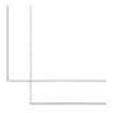
$$= (q_0^H r) q_0 + (q_1^H r) q_1 + \cdots + (q_{L-1}^H r) q_{L-1} \quad (6.34)$$

と書くことができます。 q_0, q_1, \dots, q_{N-1} が正規直交基底であることを思い出すと、これは r を q_0, q_1, \dots, q_{N-1} で表現したものと相関行列の大きな固有値に対応する固有ベクトルによる L 項で打ち切ったものと理解することができます。主成分分析（principal component analysis, PCA）^{61), 62)} と呼ばれるよく知られた解析手法となっています。つまり、主成分分析は観測ベクトルの信号部分空間への直交射影として解釈することができます。

式 (6.31) に線形観測モデルの式を代入すると

$$\hat{r} = Q_S Q_S^H A s + Q_S Q_S^H v \quad (6.35)$$

となり、右辺第 1 項が信号成分、第 2 項が雑音成分となります。これより \hat{r} の



雑音成分の全電力は

$$E[(Q_S Q_S^H v)^H Q_S Q_S^H v] = \text{tr}\{E[Q_S Q_S^H v (Q_S Q_S^H v)^H]\} \quad (6.36)$$

$$= \text{tr}\{Q_S Q_S^H E[v v^H] Q_S Q_S^H\} \quad (6.37)$$

$$= \sigma_v^2 \text{tr}\{Q_S Q_S^H\} \quad (6.38)$$

$$= L \sigma_v^2 \quad (6.39)$$

となります。

一方、元の観測ベクトル r 中の雑音成分の全電力は $E[v^H v] = N \sigma_v^2$ ですの
で、 $N \geq L$ より雑音の一部が除去されていることがわかります。

また、式 (6.28) より、 A の各列 a_0, a_1, \dots, a_{L-1} は q_0, q_1, \dots, q_{L-1} の線形
結合で表現できるので、適当な $L \times L$ の正則行列 (non-singular matrix) B
を用いて

$$A = Q_S B \quad (6.40)$$

と書けることから、信号成分は

$$Q_S Q_S^H A s = Q_S Q_S^H Q_S B s \quad (6.41)$$

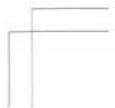
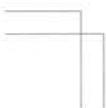
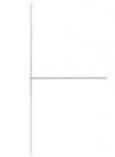
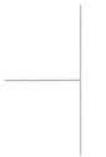
$$= Q_S B s \quad (6.42)$$

$$= A s \quad (6.43)$$

となり、信号部分空間への直交射影によって影響を受けないことが確認できます。

雑音部分空間の利用については、式 (6.29) より、 A の列ベクトルと $Q_N = [q_L \ q_{L+1} \ \cdots \ q_{N-1}]$ が直交することを利用した A についてのパラメータ推定
があります。このアプローチは雑音部分空間への直交射影を考える手法と解釈
することができるためマイナー成分分析 (minor component analysis) と呼
ぶこともありますが、単に部分空間法と呼ぶことが多いようです。

雑音部分空間を利用した到來方向推定法である MUSIC 法と、信号部分空間
を利用した ESPRIT 法について次節以降で説明します。



6.4 MUSIC 法

MUSIC 法⁶³⁾ (multiple signal classification method) は代表的な到来方向推定法であり、レイリー限界 (Rayleigh's limit)[†]を超える高い推定精度で知られています。ここでは MUSIC 法の基本的な考え方について説明します。

再び、直線センサアレーでの受信信号モデル

$$r(n, k) = \sum_{l=0}^{L-1} s_l(k) e^{j2\pi \frac{d_n \sin \theta_l}{\lambda}} + v(n, k) \quad (6.44)$$

について考えます。角度 θ の方向からの到来波に対し、各センサにおける受信信号の 0 番目のセンサでの受信信号との位相差を要素にもつベクトル

$$\mathbf{a}(\theta) = \left[e^{j2\pi \frac{d_0 \sin \theta}{\lambda}} \ e^{j2\pi \frac{d_1 \sin \theta}{\lambda}} \ \cdots \ e^{j2\pi \frac{d_{N-1} \sin \theta}{\lambda}} \right]^T \quad (6.45)$$

を導入することで、式 (6.44) の時刻 k における各センサでの受信信号をまとめたベクトル（スナップショット）は

$$\begin{aligned} r(k) &= [r(0, k) \ r(1, k) \ \cdots \ r(N-1, k)]^T \\ &= \sum_{l=0}^{L-1} s_l(k) \mathbf{a}(\theta_l) + \mathbf{v}(k) \\ &= \mathbf{A}s(k) + \mathbf{v}(k) \end{aligned} \quad (6.46)$$

と書くことができます。ただし

$$\mathbf{v}(k) = [v(0, k) \ v(1, k) \ \cdots \ v(N-1, k)]^T$$

$$\mathbf{A} = [\mathbf{a}(\theta_0) \ \mathbf{a}(\theta_1) \ \cdots \ \mathbf{a}(\theta_{L-1})]$$

$$\mathbf{s}(k) = [s_0(k) \ s_1(k) \ \cdots \ s_{L-1}(k)]^T$$

[†] プリズムや回折格子による分光器の周波数分解能の分析のために Rayleigh によって導入された規準で、長らく古典的なフーリエ変換ベースの周波数推定の分解能の限界を与えると考えられていました。

です。

なお、 $\mathbf{a}(\theta)$ は到來方向を表すパラメータ θ と各センサでの位相差の関係を対応づけるベクトルであり、 $\mathbf{a}(\theta)$ の各成分の位相差を打ち消すように各アンテナでの受信信号に位相回転を与えてそれらを足し合わせると、合成出力において対応する θ の方向に強い**指向性** (directivity) を与えることができます。複数のアンテナ素子の出力に適切な複素数の重みを乗算して合成することでアンテナアレー全体の指向性をソフトウェア的に制御することを**ビームフォーミング** (beamforming) と呼びます。

到來角 θ に応じて $\mathbf{a}(\theta)$ で定まるように各アンテナ素子での位相回転量を制御することでアンテナの指向性を走査することができるため、 $\mathbf{a}(\theta)$ は**ステアリングベクトル** (steering vector) と呼ばれます。ステアリングベクトルで構成される行列 \mathbf{A} の各列は初項が 1 の等比数列でその成分が定まっていることに注意します。このような構造をもつ行列は**ヴァンデルモンド行列** (Vandermonde matrix) と呼ばれ、各列の 2 行目の成分がすべて異なり $N \geq L$ であれば、 \mathbf{A} が列フルランクとなることが知られています。文献によってはステアリングベクトルの各要素の e の肩の数字がマイナスになっていることがあります、これは到來角をどの部分の角度で定義するかやセンサの並び順の定義が本書と異なることによって生じている相違で、本質的には同じ式と考えて構いません。

式 (6.46) は式 (6.12) と同じ形の信号モデルとなっているので、 $\mathbf{s}(k)$ と $\mathbf{v}(k)$ は無相関、 $E[\mathbf{v}(k)\mathbf{v}^H(k)] = \sigma_v^2 \mathbf{I}_N$ 、 $\mathbf{R}_s = E[\mathbf{s}(k)\mathbf{s}^H(k)]$ とすると、 $\mathbf{r}(k)$ の相關行列は

$$\begin{aligned}\mathbf{R}_r &= E[\mathbf{r}(k)\mathbf{r}^H(k)] \\ &= \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_v^2 \mathbf{I}_N\end{aligned}\tag{6.47}$$

となります。このため、 \mathbf{R}_r の N 個の固有値をその大きさにより $\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{N-1}$ とし、 $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ の N 個の固有値を $\gamma_0 \geq \gamma_1 \geq \dots \geq \gamma_{N-1}$ とすると

$$\lambda_n = \begin{cases} \gamma_n + \sigma_v^2, & n = 0, 1, \dots, L-1 \\ \sigma_v^2, & n = L, L+1, \dots, N-1 \end{cases} \quad (6.48)$$

が得られます。

\mathbf{R}_r の最小固有値 σ_v^2 に対応する $N-L$ 個の固有ベクトル $\mathbf{q}_L, \mathbf{q}_{L+1}, \dots, \mathbf{q}_{N-1}$ は雑音部分空間 $\mathcal{N}(\mathbf{A}^H)$ の基底で

$$\mathbf{q}_n^H \mathbf{A} = 0, \quad n = L, L+1, \dots, N-1 \quad (6.49)$$

を満足します。MUSIC アルゴリズムは、式 (6.49) で表される信号の到来角に 対応するステアリングベクトル $\mathbf{a}(\theta_l)$ と相関行列の最小固有値に対応した固有ベクトルが直交する性質を利用したアルゴリズムです。

具体的には、 $\mathbf{Q}_N = [\mathbf{q}_L \ \mathbf{q}_{L+1} \ \cdots \ \mathbf{q}_{N-1}]$ を用いて

$$S(\theta) = \frac{1}{\mathbf{a}^H(\theta) \mathbf{Q}_N \mathbf{Q}_N^H \mathbf{a}(\theta)} \quad (6.50)$$

を計算します。 θ の値を変化させながら式 (6.50) をプロットしたものは **MUSIC スペクトル** (MUSIC spectrum) と呼ばれます。到来波の到来角に対応した $\theta = \theta_l, l = 0, 1, \dots, L-1$ のところで式 (6.50) の分母が 0 となることから MUSIC スペクトルに鋭いピークができるため、ピークの位置から信号の到来角を推定することができます。

図 6.2 に、 -20 deg と 40 deg の角度から狭帯域信号が到来する際に、7 素子の等間隔直線アレーで受信した信号を用いて計算した MUSIC スペクトルの例を示します。信号の到来角に対応する -20 deg と 40 deg の位置に鋭いピークが確認されます。

応用によっては、 θ を走査しながら式 (6.50) の値を評価することが計算量的に難しい場合がありますが、そのような場合には式 (6.50) の分母が 0 となる θ_l を直接求めるアプローチがあり、root MUSIC 法⁶⁴⁾ と呼ばれます。

MUSIC 法では雑音部分空間を構成する固有ベクトルが最低一つは必要なため、到来波数 L がセンサ素子数 N 以上の場合には到来方向を推定できないこと

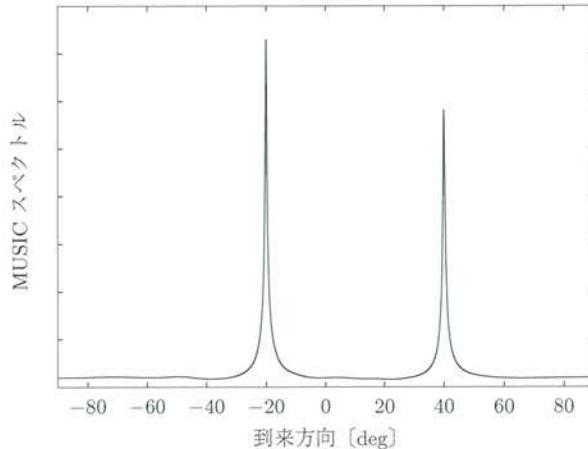


図 6.2 MUSIC スペクトルの例

に注意しましょう。また、実際には受信信号の相関行列 \mathbf{R}_r は未知であり、十分な数のスナップショット $\mathbf{r}(k)$ を用いた時間平均によるサンプル相関行列によって推定することになります。このため、到来方向に対応する θ_l に対して式 (6.50) の分母は厳密に 0 にはならず、式 (6.50) の値が発散する心配はありません。

特別な場合として、観測雑音がない状況を考えると、受信信号のサンプル相関行列は

$$\begin{aligned}\hat{\mathbf{R}}_r &= \frac{1}{K} \sum_{k=0}^{K-1} \mathbf{r}(k) \mathbf{r}^H(k) \\ &= \mathbf{A} \left(\frac{1}{K} \sum_{k=0}^{K-1} \mathbf{s}(k) \mathbf{s}^H(k) \right) \mathbf{A}^H \\ &= \mathbf{A} \hat{\mathbf{R}}_s \mathbf{A}\end{aligned}\quad (6.51)$$

で与えられます。ただし、 K はサンプル相関行列の計算に用いられるスナップショット数であり、 $\hat{\mathbf{R}}_s$ は

$$\hat{\mathbf{R}}_s = \frac{1}{K} \sum_{k=0}^{K-1} \mathbf{s}(k) \mathbf{s}^H(k) \quad (6.52)$$

で定義されます。送信信号のサンプル相関行列 $\hat{\mathbf{R}}_s$ がフルランクであれば、サンプル相関行列 $\hat{\mathbf{R}}_r$ の固有値 0 に対応する固有ベクトルは雑音部分空間を張ります。送信信号がインコヒーレントの場合には、スナップショット数が $K \geq L$ を満足すれば $\hat{\mathbf{R}}_s$ はフルランクとなるため、雑音のない場合には、原理的には到来波数と同数以上のスナップショットがあれば MUSIC 法によって到来方向推定が可能であることがわかります。

6.5 空 間 平 滑 化

ここまで MUSIC 法の説明では、送信信号の相関行列 \mathbf{R}_s やサンプル相関行列 $\hat{\mathbf{R}}_s$ がフルランクであることを仮定しましたが、到来波がコヒーレントである場合などはこの仮定が成立せず、そのまま MUSIC 法を適用してもうまく到来角を推定できません。これは一つの波源からの信号がマルチパス通信路の影響で異なる方向から同時に到来しているような状況に対応します。

具体的には、 $\alpha_0, \alpha_1, \dots, \alpha_{L-1}$ を複素の定係数として、送信信号ベクトルが

$$\mathbf{s}(k) = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{L-1} \end{bmatrix} s_0(k) \quad (6.53)$$

と書かれる場合です。この信号ベクトルの相関行列は

$$\mathbf{R}_s = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{L-1} \end{bmatrix} \begin{bmatrix} \alpha_0^* & \alpha_1^* & \cdots & \alpha_{L-1}^* \end{bmatrix} E[|s_0(k)|^2] \quad (6.54)$$

となり、明らかに \mathbf{R}_s のランクが 1 になります。このとき、到来波数が L であるにもかかわらず $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ のランクも 1 となり、受信信号ベクトルの相関行

列 \mathbf{R}_r の最小固有値 σ_v^2 に対応する固有ベクトルの張る空間の次元が $N - 1$ となってしまい、MUSIC 法によって到来方向を推定することができません。より一般に、到来波のうち二つがコヒーレントである場合も \mathbf{R}_s がフルランクにならず、同様の問題が発生します。

このような場合には、空間平滑化⁶⁵⁾と呼ばれる手法が利用されます。空間平滑化では N 素子からなる等間隔直線アレーを仮定し、それを重複する形で $P (< N)$ 素子からなる $N - P + 1$ 個のサブアレーに分割します。すなわち、0 番目のサブアレーは $\{0, 1, \dots, P-1\}$ 番目の素子からなり、1 番目のサブアレーは $\{1, 2, \dots, P\}$ 番目の素子、最後の $N - P$ 番目のサブアレーは $\{N - P, N - P + 1, \dots, N - 1\}$ 番目の素子からなる、といった具合です。

このとき、0 番目のサブアレーでの受信信号ベクトルは

$$\begin{aligned} \mathbf{r}_{0,P}(k) &= [r(0, k) \ r(1, k) \ \cdots \ r(P-1, k)]^\top \\ &= \sum_{l=0}^{L-1} s_l(k) \mathbf{a}_P(\theta_l) + \mathbf{v}_{0,P}(k) \\ &= \mathbf{A}_P \mathbf{s}(k) + \mathbf{v}_{0,P}(k) \end{aligned} \quad (6.55)$$

と書けます。ただし

$$\begin{aligned} \mathbf{v}_{0,P}(k) &= [v(0, k) \ v(1, k) \ \cdots \ v(P-1, k)]^\top \\ \mathbf{a}_P(\theta_l) &= \left[1 \ e^{j\phi_l} \ \cdots \ e^{j\phi_l(P-1)} \right]^\top \\ \phi_l &= 2\pi \frac{d \sin \theta_l}{\lambda} \\ \mathbf{A}_P &= [\mathbf{a}_P(\theta_0) \ \mathbf{a}_P(\theta_1) \ \cdots \ \mathbf{a}_P(\theta_{L-1})] \end{aligned}$$

です。

1 番目のサブアレーは 0 番目のサブアレーのすべての素子を隣に一つシフトさせることで得られますが、1 素子隣にシフトすることによる l 番目の波源からの信号に対応する受信信号の影響は $e^{j\phi_l}$ 位相を回転することに対応しますので、1 番目のサブアレーでの受信信号ベクトルは

$$\begin{aligned}
\mathbf{r}_{1,P}(k) &= [r(1,k) \ r(2,k) \ \cdots \ r(P,k)]^T \\
&= \sum_{l=0}^{L-1} s_l(k) e^{j\phi_l} \mathbf{a}_P(\theta_l) + \mathbf{v}_{1,P}(k) \\
&= \mathbf{A}_P \Phi \mathbf{s}(k) + \mathbf{v}_{1,P}(k)
\end{aligned} \tag{6.56}$$

と書くことができます。ここで

$$\begin{aligned}
\mathbf{v}_{1,P}(k) &= [v(1,k) \ v(2,k) \ \cdots \ v(P,k)]^T, \\
\Phi &= \text{diag}[e^{j\phi_0} \ e^{j\phi_1} \ \cdots \ e^{j\phi_{L-1}}]
\end{aligned} \tag{6.57}$$

です。

同様に考えると m 番目のサブアレーでの受信信号ベクトルは

$$\begin{aligned}
\mathbf{r}_{m,P}(k) &= [r(m,k) \ r(m+1,k) \ \cdots \ r(m+P-1,k)]^T \\
&= \mathbf{A}_P \Phi^m \mathbf{s}(k) + \mathbf{v}_{m,P}(k)
\end{aligned} \tag{6.58}$$

と書けます。ただし

$$\mathbf{v}_{m,P}(k) = [v(m,k) \ v(m+1,k) \ \cdots \ v(m+P-1,k)]^T$$

です。これより、 $\mathbf{r}_m(k)$ の相関行列は

$$\begin{aligned}
\mathbf{R}_{\mathbf{r}_m} &= E[\mathbf{r}_m(k) \mathbf{r}_m^H(k)] \\
&= \mathbf{A}_P \Phi^m \mathbf{R}_s(\Phi^m)^H \mathbf{A}_P^H + \sigma_v^2 \mathbf{I}_P
\end{aligned} \tag{6.59}$$

となります。

$\mathbf{R}_{\mathbf{r}_m}$ に直接 MUSIC 法を適用しようとしても \mathbf{R}_s がフルランクでなければうまくいきませんが、空間平滑化では各サブアレーの相関行列の標本平均で定義される行列

$$\bar{\mathbf{R}}_r = \frac{1}{N-P+1} \sum_{m=0}^{N-P+1} \mathbf{R}_{\mathbf{r}_m} \tag{6.60}$$

を考えます。これは空間平滑化された相関行列と呼ばれ

$$\bar{\mathbf{R}}_r = \mathbf{A}_P \bar{\mathbf{R}}_s \mathbf{A}_P^H + \sigma_v^2 \mathbf{I}_P \quad (6.61)$$

と書くことができます。ただし

$$\bar{\mathbf{R}}_s = \frac{1}{N-P+1} \sum_{m=0}^{N-P+1} \Phi^m \mathbf{R}_s (\Phi^m)^H \quad (6.62)$$

です。

ここで重要な点は、 $\bar{\mathbf{R}}_r$ が全素子での受信信号 $\mathbf{r}(k)$ の相関行列 \mathbf{R}_r と同じ形で与えられ、かつ、サブアレーの数 $N - P + 1$ が到來波数 L 以上であれば、波源からの信号がコヒーレントであっても $\bar{\mathbf{R}}_s$ がフルランクになるため、 $\bar{\mathbf{R}}_r$ に對して MUSIC 法を適用することでコヒーレントな信号の到來方向推定が可能であるということです。

$\bar{\mathbf{R}}_s$ がフルランクになることはつきのように確かめられます。

Φ が式 (6.57) で定義される対角行列であることに注意すると $\Phi^H = \Phi^{-1}$ となるので、 $\bar{\mathbf{R}}_s$ は

$$\bar{\mathbf{R}}_s = \frac{1}{N-P+1} \begin{bmatrix} \mathbf{I}_P & \Phi & \cdots & \Phi^{N-P} \end{bmatrix} \begin{bmatrix} \mathbf{R}_s & & & \\ & \ddots & & \\ & & \mathbf{R}_s & \\ & & & \Phi^{-(N-P)} \end{bmatrix} \begin{bmatrix} \mathbf{I}_P \\ \Phi^{-1} \\ \vdots \\ \Phi^{-(N-P)} \end{bmatrix} \quad (6.63)$$

と書くことができます。さらに $\bar{\mathbf{R}}_s$ は

$$\frac{1}{N-P+1} \mathbf{R}_s = \mathbf{C} \mathbf{C}^H \quad (6.64)$$

を満たす $1/(N-P+1)\mathbf{R}_s$ のエルミート平方根行列 (Hermitian square root matrix) $\mathbf{C} \in \mathbb{C}^{L \times L}$ を用いて[†]

[†] \mathbf{R}_s は相関行列なので非負定値（半正定値）であり、平方根行列が一意に定まります。

$$\bar{\mathbf{R}}_s = \mathbf{E}\mathbf{E}^H \quad (6.65)$$

と書くことができます。ただし

$$\mathbf{E} = [\mathbf{C} \quad \Phi\mathbf{C} \quad \dots \quad \Phi^{N-P}\mathbf{C}] \in \mathbb{C}^{L \times (N-P+1)L} \quad (6.66)$$

です。

式 (6.65) より $\bar{\mathbf{R}}_s$ のランクは \mathbf{E} のランクと等しいので、以下では \mathbf{E} がサブアレー数の条件の下で行フルランク L となることを示します。行列はその列や行を入れ替えてランクが変化しないことに注意すると

$$\text{rank}\{\mathbf{E}\} = \text{rank} \left\{ \begin{bmatrix} c_{0,0}\phi_0 & c_{0,1}\phi_0 & \cdots & c_{0,L-1}\phi_0 \\ c_{1,0}\phi_1 & c_{1,1}\phi_1 & \cdots & c_{1,L-1}\phi_1 \\ \vdots & \vdots & \ddots & \vdots \\ c_{L-1,0}\phi_{L-1} & c_{L-1,1}\phi_{L-1} & \cdots & c_{L-1,L-1}\phi_{L-1} \end{bmatrix} \right\} \quad (6.67)$$

となります。ただし、 $c_{i,j}$ は \mathbf{C} の (i, j) 成分であり

$$\phi_l = [1 \quad e^{j\phi_l} \quad \dots \quad e^{j\phi_l(N-P)}] \quad (6.68)$$

です。ここで、 \mathbf{C} の各行は少なくとも一つの要素が 0 ではないこと[†]、から、 $\{\phi_0, \phi_1, \dots, \phi_{L-1}\}$ が 1 次独立なら \mathbf{E} がフルランクになります。さらに、式 (6.68) より、 ϕ_l はヴァンデルモンド行列の各列の構造をもつので $N - P + 1 \geq L$ を満足すれば、 $\{\phi_0, \phi_1, \dots, \phi_{L-1}\}$ が 1 次独立となります。以上より、サブアレーの数が $N - P + 1 \geq L$ を満足すれば $\bar{\mathbf{R}}_s$ がフルランクとなることが確認できました。

空間平滑化は少し理解しにくいかもしれません、直観的には、異なる到来方向をもつ信号は受信センサを空間的にずらしたときに異なる影響を受けるた

[†] もしある行の成分がすべて 0 であればそれに対応する信号の受信電力が 0 となるので、 L 個の波源から信号が到来しているという設定に矛盾します。

め、それらを時間軸におけるスナップショット間の信号の異なる変動とみなして利用することで、コヒーレントな信号をインコヒーレントな信号に変換していると解釈することができます。

空間平滑化を前処理に用いた MUSIC 法では、空間平滑化における要請 $N - P + 1 \geq L$ と MUSIC 法における要請 $P > L$ の両方を満足する必要があります。このため、アンテナアレー全体の素子数は到来波数の 2 倍以上、すなわち、 $N \geq 2L$ であることが必要です。通常の MUSIC 法の条件 $N > L + 1$ と比べると、実効的なアレーの自由度がおよそ半分になっていることがわかります。より効率的な空間平滑化法として $N \geq 3L/2$ の素子数のみを必要とする空間平滑化法も提案されているので、興味のある方は文献66), 67) などを参照されるとよいでしょう。

6.6 ESPRIT 法

ESPRIT 法⁶⁸⁾ (estimation of signal parameters via rotational invariance techniques method) は MUSIC 法と並んでよく知られた到来方向推定のためのアルゴリズムで、部分空間法および電波干渉計 (radio interferometer) の原理に基づいた手法です。ESPRIT 法では二つの同一のサブアレーで構成されるセンサアレーでの受信信号を利用します。二つのサブアレーが同一の配置をもつならば任意の配置のアレーが利用できますが、ここでは簡単のため、等間隔直線アレーを用いた場合について説明します。

N 素子からなる素子間隔 d の等間隔直線アレーに L 波のインコヒーレントな狭帯域信号が到来しているとし、空間平滑化で考えたのと同様に、重複する形で $N - 1$ 素子からなる二つのサブアレーを考えます。つまり、素子 $\{0, 1, \dots, N - 2\}$ で構成されるサブアレーをサブアレー 0 とし、素子 $\{1, 2, \dots, N - 1\}$ で構成されるサブアレーをサブアレー 1 とします。これは空間平滑化で考えた信号モデルで $P = N - 1$ とした場合に対応することに注意すると、サブアレー 0 と 1 での受信信号ベクトルはそれぞれ次のように書けます。

$$\mathbf{r}_{0,N-1}(k) = \mathbf{A}_{N-1}\mathbf{s}(k) + \mathbf{v}_{0,N-1}(k) \quad (6.69)$$

$$\mathbf{r}_{1,N-1}(k) = \mathbf{A}_{N-1}\Phi\mathbf{s}(k) + \mathbf{v}_{1,N-1}(k) \quad (6.70)$$

一方、サブアレー 0 での受信信号ベクトルは、等間隔直線アレーのすべての素子からの受信信号ベクトル

$$\mathbf{r}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{v}(k) \quad (6.71)$$

の最初の $N - 1$ 個の成分で構成されることから

$$\mathbf{r}_{0,N-1}(k) = \mathbf{J}_0\mathbf{r}(k) \quad (6.72)$$

と書くこともできます。ここで、行列 \mathbf{J}_0 はベクトルに左から乗算することでベクトルの最初の $N - 1$ 個の成分を抽出する働きをもち

$$\mathbf{J}_0 = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(N-1) \times N} \quad (6.73)$$

で定義されます。

同様に、サブアレー 1 での受信信号ベクトルは、等間隔直線アレーのすべての素子からの受信信号ベクトル $\mathbf{r}(k)$ を用いて

$$\mathbf{r}_{1,N-1}(k) = \mathbf{J}_1\mathbf{r}(k) \quad (6.74)$$

と書くことができます。ここで、行列 \mathbf{J}_1 はベクトルに左から乗算することでベクトルの最後の $N - 1$ 個の成分を抽出する働きをもち

$$\mathbf{J}_1 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(N-1) \times N} \quad (6.75)$$

で定義されます。

式 (6.72), (6.74) より, サブアレー 0, 1 におけるステアリング行列はそれぞれ \mathbf{A} を用いて

$$\mathbf{A}_{N-1} = \mathbf{J}_0 \mathbf{A} \quad (6.76)$$

$$\mathbf{A}_{N-1} \Phi = \mathbf{J}_1 \mathbf{A} \quad (6.77)$$

と書くことができるので

$$\mathbf{J}_0 \mathbf{A} \Phi = \mathbf{J}_1 \mathbf{A} \quad (6.78)$$

なる関係が得られます。

Φ が位相回転を表す行列であることから, この関係は**回転不变性** (rotational invariance) と呼ばれ, これが ESPRIT の名前に含まれていることからもわかるように, ESPRIT 法で中心的な役割を果たします。すなわち, ESPRIT 法では Φ にすべての到来波の到来角の情報が含まれていることに着目し, ステアリング行列 \mathbf{A} ではなく Φ を推定することで到来角を求めます。このことは, MUSIC 法がステアリング行列の構造と, 信号部分空間と雑音部分空間の直交性を利用してステアリング行列 \mathbf{A} そのものを推定していたのと対照的です。

関係式 (6.78) を用いると, 例えば最小 2 乗法によって

$$\Phi = \{(\mathbf{J}_0 \mathbf{A})^\text{H} \mathbf{J}_0 \mathbf{A}\}^{-1} (\mathbf{J}_0 \mathbf{A})^\text{H} \mathbf{J}_1 \mathbf{A} \quad (6.79)$$

$$= (\mathbf{A}^\text{H} \mathbf{J}_0^\text{T} \mathbf{J}_0 \mathbf{A})^{-1} \mathbf{A}^\text{H} \mathbf{J}_0^\text{T} \mathbf{J}_1 \mathbf{A} \quad (6.80)$$

と Φ を推定できますが, \mathbf{A} は未知ですのでもちろんこの方法は使えません。そこで, 部分空間法の「 \mathbf{A} を直接求めることはできないが, その列ベクトルの張る空間なら信号部分空間として相関行列から求めることができる」という性質を利用します。

すべてのアンテナ素子での受信信号ベクトル $\mathbf{r}(k)$ の相関行列 \mathbf{R}_r を求めて, その大きなほうから L 個の固有値に対応する固有ベクトルを列ベクトルにもつ行列 \mathbf{Q}_S を得ると, ある正則な行列 $\mathbf{T} \in \mathbb{C}^{L \times L}$ を用いて

$$Q_S = AT \quad (6.81)$$

と書くことができます。これを式 (6.78) に代入することで

$$J_0 Q_S T^{-1} \Phi = J_1 Q_S T^{-1} \quad (6.82)$$

となり

$$J_0 Q_S \Psi = J_1 Q_S \quad (6.83)$$

が得られます。ただし

$$\Psi = T^{-1} \Phi T \quad (6.84)$$

です。

式 (6.83) は元の回転不变式 (6.78) とよく似た形をしていますが、 Q_S は相関行列 R_r から実際に求めることができます。この方程式を解くことで Ψ を得ることができます。一方、 Ψ から Φ を求める際に T が未知であることが問題になるように思えますが、 Φ が対角行列であることから式 (6.84) は Ψ の固有方程式にはかならず、 Ψ を固有値分解して固有値を求めて Φ が得られます。

Φ の l 番目の対角成分は、到來波の波長を λ 、 l 番目の到來波の到來角を θ_l として

$$e^{j\phi_l} = e^{j2\pi \frac{d \sin \theta_l}{\lambda}} \quad (6.85)$$

と書けるので、 Ψ の固有値を ψ_l 、 $l = 0, 1, \dots, L - 1$ とすると、到來角は

$$\theta_l = \sin^{-1} \left(\frac{\lambda}{2\pi d} \arg(\psi_l) \right) \quad (6.86)$$

と推定されます。ただし、 ψ_l から到來角 θ_l を推定するためには、前述のように素子間隔 d が $\lambda/2$ 以下である必要があることに注意します。

式 (6.83) を用いて Ψ を求める際には、最小 2 乗法によって

$$\Psi = \{(J_0 Q_S)^H J_0 Q_S\}^{-1} (J_0 Q_S)^H J_1 Q_S \quad (6.87)$$

$$= (Q_S^H J_0^T J_0 Q_S)^{-1} Q_S^H J_0^T J_1 Q_S \quad (6.88)$$

とする方法がよく用いられます。これを用いた ESPRIT 法は LS-ESPRIT と呼ばれます。最小 2 乗法以外の方法で Ψ を求めることで、LS-ESPRIT よりも良好な推定精度が得られる方法として TLS-ESPRIT があります。興味のある方は文献68), 69) などを参照してください。

ESPRIT 法で推定可能な到来波数は、式 (6.88) の $Q_S^H J_0^T J_0 Q_S$ が正則となるための条件からわかるように、 N 素子の等間隔直線アレーに対して $N - 1$ 波までとなっており、これは MUSIC 法の条件に一致します。一般に ESPRIT 法で推定可能な到来波数はサブアレーのセンサ素子数と同じ数までなので、本節で仮定したようなサブアレー間でセンサが重複する配置をとらなければ、MUSIC 法に比べておよそ 2 倍のセンサ素子数が必要になります⁶⁸⁾。

6.7 KR 積拡張アレー処理による到来方向推定

MUSIC 法を含む従来の古典的な到来方向推定法では、到来波数がセンサ素子数より少ない場合にしか到来角を推定することができませんでしたが、Khatri-Rao (KR) 積 (Khatri-Rao product) を用いた拡張アレー処理と呼ばれる手法を用いることで、 N 素子の等間隔直線アレーの場合には $2N - 2$ 個までの到来波に対して、配置が最適化された N 素子アレーでは最大 $N(N - 1)$ 個の到来波に対して、到来方向を推定可能な手法が 2010 年頃に提案されました^{70)~72)}。ここでは、KR 積とそれに関連する演算について復習した後、KR 積拡張アレー処理の考え方について説明し、到来波の擬定常性 (quasi-stationarity) を仮定することで KR 積拡張アレー処理後の信号に MUSIC 法を適用する手法 (KR-MUSIC 法) を紹介します。ここで紹介するアルゴリズムのほかに、KR 積拡張アレー処理後に ESPRIT 法を適用する手法⁷³⁾ も提案されています。

6.7.1 KR 積とクロネッカー積、ベクトル化

KR 積拡張アレー処理を理解するためには、まず、KR 積、クロネッカー積 (Kronecker product)、ベクトル化 (vectorization) の演算の定義とそれらの関係を押さえておく必要があります⁵⁾。

定義 6.1 (クロネッカー積) 行列 $A = [a_1 \ a_2 \ \cdots \ a_n] \in \mathbb{C}^{m \times n}$ と $B = [b_1 \ b_2 \ \cdots \ b_q] \in \mathbb{C}^{p \times q}$ のクロネッcker 積はつぎで定義される。

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \cdots & a_{m,n}B \end{bmatrix} \quad (6.89)$$

ただし、 $a_i = [a_{1,i} \ a_{2,i} \ \cdots \ a_{m,i}]^T$, $i = 1, 2, \dots, n$, $b_j = [b_{1,j} \ b_{2,j} \ \cdots \ b_{p,j}]^T$, $j = 1, 2, \dots, q$ である。

定義 6.2 (KR 積) 行列 $X = [x_1 \ x_2 \ \cdots \ x_k] \in \mathbb{C}^{n \times k}$ と $Y = [y_1 \ y_2 \ \cdots \ y_k] \in \mathbb{C}^{m \times k}$ の KR 積はつぎで定義される。

$$X \odot Y = [x_1 \otimes y_1 \ x_2 \otimes y_2 \ \cdots \ x_k \otimes y_k] \quad (6.90)$$

KR 積の定義中に現れる二つのベクトル間のクロネッcker 積の性質を見るために、ベクトル $x = [x_1 \ x_2 \ \cdots \ x_n]^T \in \mathbb{C}^n$ と $y = [y_1 \ y_2 \ \cdots \ y_m]^T \in \mathbb{C}^m$ のクロネッcker 積を成分で書き下してみると

$$x \otimes y = \begin{bmatrix} x_1y \\ x_2y \\ \vdots \\ x_ny \end{bmatrix}$$

$$= \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_m & \cdots & x_ny_1 & x_ny_2 & \cdots & x_ny_m \end{bmatrix}^T \quad (6.91)$$

となります。ここで

$$yx^T = \begin{bmatrix} x_1y_1 & x_2y_1 & \cdots & x_ny_1 \\ x_1y_2 & x_2y_2 & \cdots & x_ny_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1y_m & x_2y_m & \cdots & x_ny_m \end{bmatrix} \quad (6.92)$$

と比較すると

$$\mathbf{x} \otimes \mathbf{y} = \text{vec}(yx^T) \quad (6.93)$$

と書けることがわかります。ただし、 $\text{vec}(\cdot)$ は行列のベクトル化の操作を表し

$$\text{vec}(\mathbf{X}) = [x_1^T \ x_2^T \ \cdots \ x_k^T]^T \in \mathbb{C}^{nk} \quad (6.94)$$

で定義されます。その他のクロネッカーリングの性質については付録 A.2 節を参照してください。

つぎに述べる KR 積拡張アレー処理では、ここで説明したのと逆の流れで KR 積を用いた受信信号モデルが構成されます。

6.7.2 KR 積拡張アレー処理

KR 積拡張アレー処理では物理的なセンサ素子数よりも多くの到来波の到来方向推定が可能ですが、従来の MUSIC 法などに比べて厳しい適用条件があります。その一つが、到来信号がたがいに無相関であることです。このとき、送信信号ベクトル $\mathbf{s}(k)$ の相關行列 $\mathbf{R}_s = E[\mathbf{s}(k)\mathbf{s}^H(k)]$ は対角行列

$$\mathbf{R}_s = \text{diag}[\sigma_{s,0}^2 \ \sigma_{s,1}^2 \ \cdots \ \sigma_{s,L-1}^2] \quad (6.95)$$

となります。ただし、 $\sigma_{s,l}^2 = E[|s_l(k)|^2]$, $l = 0, 1, \dots, L-1$ です。したがって、

受信信号ベクトル $\mathbf{r}(k)$ の相関行列は

$$\begin{aligned}\mathbf{R}_r &= \mathbf{A} \mathbf{R}_s \mathbf{A}^H + \sigma_v^2 \mathbf{I}_N \\ &= \sum_{l=0}^{L-1} \sigma_{s,l}^2 \mathbf{a}(\theta_l) \mathbf{a}^H(\theta_l) + \sigma_v^2 \mathbf{I}_N\end{aligned}\quad (6.96)$$

となります。

MUSIC 法では \mathbf{R}_r の固有値分解を考えましたが、KR 積拡張アレー処理では \mathbf{R}_r をベクトル化することで新たな受信信号モデルを得ることを考えます。すなわち

$$\begin{aligned}\tilde{\mathbf{r}} &\equiv \text{vec}(\mathbf{R}_r) \\ &= \text{vec} \left(\sum_{l=0}^{L-1} \sigma_{s,l}^2 \mathbf{a}(\theta_l) \mathbf{a}^H(\theta_l) \right) + \text{vec}(\sigma_v^2 \mathbf{I}_N) \\ &= \sum_{l=0}^{L-1} \sigma_{s,l}^2 \text{vec} (\mathbf{a}(\theta_l) \mathbf{a}^H(\theta_l)) + \sigma_v^2 \tilde{\mathbf{i}}_N \\ &= \sum_{l=0}^{L-1} \sigma_{s,l}^2 \mathbf{a}^*(\theta_l) \otimes \mathbf{a}(\theta_l) + \sigma_v^2 \tilde{\mathbf{i}}_N \\ &= (\mathbf{A}^* \odot \mathbf{A}) \mathbf{p} + \sigma_v^2 \tilde{\mathbf{i}}_N\end{aligned}\quad (6.97)$$

を考えます。ただし、 $\tilde{\mathbf{i}}_N = \text{vec}(\mathbf{I}_N)$ 、 $\mathbf{p} = [\sigma_{s,0}^2 \ \sigma_{s,1}^2 \ \cdots \ \sigma_{s,L-1}^2]^T$ です。2 行目から 3 行目への変形では vec 演算が線形性をもつことを、3~5 行目への式変形では 6.7.1 項の KR 積とクロネッカーリー積、ベクトル化の演算の間の関係を、それぞれ利用しています。

式 (6.97) を式 (6.46) と比較し、 $\mathbf{A}^* \odot \mathbf{A}$ を式 (6.46) の \mathbf{A} に対応する行列、 \mathbf{p} を送信信号ベクトル、 $\sigma_v^2 \tilde{\mathbf{i}}_N$ を観測雑音ベクトルと見ることで、 $\tilde{\mathbf{r}}$ を新たな受信信号ベクトル（スナップショット）とみなすことができます。以下では、これを KR 積拡張受信信号と呼びます。

ここで重要なことは、KR 積拡張アレー処理前後での受信信号ベクトルの次元に注目すると、 $\mathbf{r}(k) \in \mathbb{C}^N$ に対して、 $\tilde{\mathbf{r}} \in \mathbb{C}^{N^2}$ となっていて、見かけ上、

センサ素子数、すなわち、アレーの自由度が N から N^2 に増加していることです。実際には、ステアリングベクトルで構成される元の行列 \mathbf{A} の n 番目、 $n = 0, 1, \dots, N - 1$ の行ベクトルが

$$\left[e^{j2\pi \frac{d_n \sin \theta_0}{\lambda}} \ e^{j2\pi \frac{d_n \sin \theta_1}{\lambda}} \ \dots \ e^{j2\pi \frac{d_n \sin \theta_{L-1}}{\lambda}} \right] \quad (6.98)$$

であるのに対し、KR 積拡張アレー処理後の行列 $\mathbf{A}^* \odot \mathbf{A}$ の $iN + j$ 番目、 $i, j = 0, 1, \dots, N - 1$ の行ベクトルは

$$\left[e^{j2\pi \frac{(d_j - d_i) \sin \theta_0}{\lambda}} \ e^{j2\pi \frac{(d_j - d_i) \sin \theta_1}{\lambda}} \ \dots \ e^{j2\pi \frac{(d_j - d_i) \sin \theta_{L-1}}{\lambda}} \right] \quad (6.99)$$

となり、KR 積拡張受信信号におけるアレーの自由度がいくつになるかは、 $d_j - d_i$, $i, j = 0, 1, \dots, N - 1$ がいくつの異なる値をとるかに依存します。

このように、センサ素子の位置が d_n , $n = 0, 1, \dots, N - 1$ で与えられる元の直線アレーに対し、センサの位置が $d_j - d_i$, $i, j = 0, 1, \dots, N - 1$ の相異なる値によって定まるアレーは**差分アレー** (difference co-array) と呼ばれます。差分アレーの自由度は元のアレーの素子配置に依存し、元のアレーとして N 素子の直線アレーを用いたときに、その差分アレーで達成可能な最大の自由度は $N(N - 1) + 1$ であることが知られています⁷²⁾。また、元のアレーが等間隔の直線アレーの場合にはその差分アレーの自由度は $2N - 1$ となり、元のアレーが 2 レベルの**入れ子型アレー**⁷²⁾ (nested array) の場合は、 N が偶数のときは $(N^2 - 2)/2 + N$ 、奇数のときは $(N^2 - 1)/2 + N$ となります。

以上より、相関行列のベクトル化操作によって得られる KR 積拡張受信信号を用いることで、実際のセンサ素子数以上のアレーの自由度を獲得することが可能となり、これによってセンサ素子数以上の到来波の推定が可能となります。

6.7.3 KR-MUSIC 法

つぎに、KR 積拡張受信信号に対して MUSIC 法を適用することで到来方向推定を行うことを考えます。MUSIC 法では複数のスナップショットを用いてサンプル相関行列を計算し、その固有値分解を行います。一方、KR 積拡張受

信信号 $\tilde{\mathbf{r}}$ は通常のアレー信号処理における一つのスナップショットに対応するため、受信信号から複数の KR 積拡張受信信号 $\tilde{\mathbf{r}}$ を生成してサンプル相関行列に相当する行列を計算することが考えられますが、MUSIC 法やその他の多くの信号処理で想定されるように到來信号が定常であることを仮定すると、 $\tilde{\mathbf{r}}$ が時間的に変化しないため MUSIC 法を適用可能なサンプル相関行列が得られません。

このため、KR 積拡張アレー処理を利用した MUSIC 法では、到來信号の電力がフレーム (frame) と呼ばれる時間区間内では一定で、異なるフレーム間では電力が変化することが仮定されます。この性質は擬定常 (quasi-stationary) と呼ばれます。

m 番目、 $m = 0, 1, \dots, M - 1$ のフレームにおける受信信号の相関行列を $\mathbf{R}_{r,m}$ とし、それぞれ KR 積拡張処理を行って得られたベクトルを並べることでデータ行列 (data matrix)

$$\begin{aligned} \mathbf{Y} &\equiv [\text{vec}(\mathbf{R}_{r,0}) \ \text{vec}(\mathbf{R}_{r,1}) \ \cdots \ \text{vec}(\mathbf{R}_{r,M-1})] \\ &= (\mathbf{A}^* \odot \mathbf{A}) \mathbf{\Pi} + \sigma^2 \tilde{\mathbf{I}}_N \mathbf{1}_M^\top \end{aligned} \quad (6.100)$$

を得ます。ここで、 $\mathbf{\Pi} = [p_0 \ p_1 \ \cdots \ p_{M-1}]$ 、 $\mathbf{1}_M = [1 \ 1 \ \cdots \ 1]^\top \in \mathbb{R}^M$ です。ただし、 $p_m, m = 0, 1, \dots, M - 1$ は m 番目のフレームにおける各波源からの信号の電力ベクトルです。

サンプル相関行列に対応する行列 (の定数倍) は $\mathbf{Y}\mathbf{Y}^\top$ によって得られますが、式 (6.100) で観測雑音に対応する右辺第 2 項は確定的な行列なので、サンプル相関行列を求める前に $\mathbf{1}_M$ の直交補空間への射影行列 $\mathbf{P}_{\mathbf{1}_M}^\perp = \mathbf{I}_M - \frac{1}{M} \mathbf{1}_M \mathbf{1}_M^\top$ を式 (6.100) の両辺に右から乗算することで、雑音のないデータ行列

$$\begin{aligned} \tilde{\mathbf{Y}} &\equiv \mathbf{Y} \mathbf{P}_{\mathbf{1}_M}^\perp \\ &= (\mathbf{A}^* \odot \mathbf{A}) \mathbf{\Pi} \mathbf{P}_{\mathbf{1}_M}^\perp \end{aligned} \quad (6.101)$$

が得られます。ここで、 $[\mathbf{\Pi}^\top, \mathbf{1}_M] \in \mathbb{R}^{M \times (L+1)}$ が列フルランクであれば、 $\text{rank}(\mathbf{\Pi} \mathbf{P}_{\mathbf{1}_M}^\perp) = \text{rank}(\mathbf{\Pi}) = L$ となり、雑音除去によって信号部分のランク

が落ちることはありません⁷⁰⁾。このとき、 $M > L$ であることに注意します。

行列 $\tilde{\mathbf{Y}}$ の各列は雑音がないときのスナップショットに対応するため、サンプル相關行列に対応する行列 $\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^H$ の固有値分解を行うことで MUSIC 法により到来角が推定できますが、通常の MUSIC 法にも適用可能な他の方法として、雑音除去後のデータ行列 $\tilde{\mathbf{Y}}$ の特異値分解

$$\tilde{\mathbf{Y}} = [\mathbf{U}_S \quad \mathbf{U}_N] \begin{bmatrix} \Sigma_S & \mathbf{0}_{L \times (N^2-L)} \\ \mathbf{0}_{(N^2-L) \times L} & \mathbf{0}_{(N^2-L) \times (N^2-L)} \end{bmatrix} \begin{bmatrix} \mathbf{V}_S^H \\ \mathbf{V}_N^H \end{bmatrix}$$

を計算するアプローチがあります。

行列 $\tilde{\mathbf{Y}}$ の特異値 0 に対応する左特異ベクトル (left-singular vector) からなる行列 $\mathbf{U}_N \in \mathbb{C}^{N^2 \times (N^2-L)}$ は雑音部分空間を張るので、KR 積拡張受信信号におけるステアリングベクトル ($\mathbf{A}^* \odot \mathbf{A}$ の列ベクトル) を用いて計算される MUSIC スペクトル

$$S_{kr}(\theta) = \frac{1}{(\mathbf{a}^*(\theta) \otimes \mathbf{a}(\theta))^H \mathbf{U}_N \mathbf{U}_N^H (\mathbf{a}^*(\theta) \otimes \mathbf{a}(\theta))} \quad (6.102)$$

を評価することで到来角を推定できます。

ここでは到来信号の擬定常性を仮定することで MUSIC 法を適用できるようにしましたが、コヒーレントな波源からの信号の到来方向推定を行うために提案された前述の空間平滑化を利用することで、到来信号の擬定常性を仮定することなく MUSIC 法によって到来方向を推定することができます。ただし、空間平滑化は等間隔直線アレーに対して有効な手法であるため、差分アレーが等間隔直線アレーとなるように、元のアレーとして入れ子型アレーなどを用いる必要があります。また、空間平滑化を利用することで、推定可能な到来波数が擬定常性を仮定する KR-MUSIC 法の半分になってしまうという欠点があります。空間平滑化以外の方法でコヒーレントな到来波を扱うことができる手法として、KR 積拡張受信信号のモデルを一般化したクロネッカー積による受信信号モデルを利用する手法も提案されています⁷⁴⁾。

6.8 む す び

本章では部分空間法や MUSIC 法の基本的な考え方について説明しました。MUSIC 法の詳細や性能解析、その他の古典的な到來方向推定アルゴリズムについては、文献69), 75)~78)などを参照してください。また、到來方向推定は圧縮センシングと密接な関係があることが知られており、文献79), 80)などで議論されています。本章では部分空間法の応用として到來方向推定のみを取り上げましたが、よく知られた他の応用例として、既知信号に対する入出力関係を用いずにシステムのインパルス応答を推定する**ブラインド同定法**^{81), 82)}(blind identification)があり、**マルチチャネル信号処理**(multichannel signal processing)と呼ばれるアプローチで信号の周期定常性がうまく利用されています。周期定常性と部分空間法の組合せは、広帯域信号の到來方向推定⁸³⁾などでも利用されます。周期定常性に興味のある方は、2次の統計量を用いたその他のブラインド同定法^{84), 85)}や、周期定常性を利用した信号処理に関する成書¹⁰⁾などを参照されるとよいでしょう。

7

状 態 推 定

Next SIP

通信や信号処理の多くの問題では、時系列データのような各時刻で得られる観測に基づいて逐次的に推定を行う必要があります。このような問題に対する強力な手法が、時系列をマルコフ性をもつ確率過程と仮定することで得られるモデル（状態空間モデル）に基づいて**状態推定**（state estimation）を行うアプローチです。本章では、状態空間モデルを導入し、状態推定の基本的な考え方について説明した後、代表的な状態推定法である粒子フィルタとカルマンフィルタについて説明します。

7.1 状態空間モデル

未知の興味のある離散時間確率過程（状態（state）と呼びます）を X_1, X_2, \dots, X_T とし、それぞれに対する観測の確率過程を Y_1, Y_2, \dots, Y_T とします。状態 X_1, X_2, \dots, X_T および観測 Y_1, Y_2, \dots, Y_T の同時分布の（一般化された）確率密度関数 $p(x_1, x_2, \dots, x_T, y_1, y_2, \dots, y_T)$ は、確率の基本法則（乗法定理）を繰り返し用いることで以下のように変形することができます。

$$\begin{aligned}
 & p(x_1, x_2, \dots, x_T, y_1, y_2, \dots, y_T) \\
 &= p(y_T | x_1, x_2, \dots, x_T, y_1, y_2, \dots, y_{T-1}) \\
 &\quad \cdot p(x_1, x_2, \dots, x_T, y_1, y_2, \dots, y_{T-1}) \\
 &= p(y_T | x_{1:T}, y_{1:T-1}) p(x_{1:T}, y_{1:T-1}) \\
 &= p(y_T | x_{1:T}, y_{1:T-1}) p(x_T | x_{1:T-1}, y_{1:T-1}) p(x_{1:T-1}, y_{1:T-1})
 \end{aligned}$$

$$\begin{aligned}
 &= p(y_T | x_{1:T}, y_{1:T-1}) p(x_T | x_{1:T-1}, y_{1:T-1}) \\
 &\quad \cdot p(y_{T-1} | x_{1:T-1}, y_{1:T-2}) p(x_{T-1} | x_{1:T-2}, y_{1:T-2}) p(x_{1:T-2}, y_{1:T-2}) \\
 &\quad \vdots \\
 &= \prod_{t=1}^T p(y_t | x_{1:t}, y_{1:t-1}) p(x_t | x_{1:t-1}, y_{1:t-1})
 \end{aligned} \tag{7.1}$$

ただし $x_{1:T}$ は x_1, x_2, \dots, x_T を意味し、表記を簡単にするために $p(y_1 | x_{1:1}, y_{1:0}) = p(y_1 | x_1)$ および $p(x_1 | x_{1:0}, y_{1:0}) = p(x_1)$ と定義します。

ここまで変形では何の仮定も導入しておらず、式 (7.1) は単に同時分布の密度関数の別表現を与えたにすぎません。ここで、状態と観測のマルコフ性 (Markov property) を仮定することによって、状態推定がきわめて効率的に行える確率モデルが構築されます。具体的には、 X_t は X_{t-1} にのみ依存し、 Y_t は X_t にのみ依存すると仮定すると

$$p(x_t | x_{1:t-1}, y_{1:t-1}) = p(x_t | x_{t-1}) \tag{7.2}$$

$$p(y_t | x_{1:t}, y_{1:t-1}) = p(y_t | x_t) \tag{7.3}$$

となり、式 (7.1) は

$$p(x_{1:T}, y_{1:T}) = \prod_{t=1}^T p(y_t | x_t) p(x_t | x_{t-1}) \tag{7.4}$$

と書けます。ただし、表記を簡単にするために $p(x_1 | x_0) = p(x_1)$ と定義しています。

同時分布の密度関数が式 (7.4) のように因数分解されることを利用して効率的に状態 X_t の周辺事後確率密度関数を計算することが可能です。粒子フィルタやカルマンフィルタはその代表的なアルゴリズムとして知られています。

式 (7.2) および式 (7.3) の仮定に注目すると、状態 X_t および観測 Y_t がどのように生成されるかについてのモデルとなっており、それぞれ、**システムモデル** (system model)、**観測モデル** (measurement model) と呼ばれます。また、こ

これらをまとめて一般(化)状態空間モデル(generalized state space model)といいます⁸⁷⁾。

定義 7.1 (一般状態空間モデル)

$x_t \sim p(x_t|x_{t-1})$: システムモデル

$y_t \sim p(y_t|x_t)$: 観測モデル

式(7.4)の同時分布の確率密度関数をベイジアンネットワークで表現すると図7.1に示すような鎖状の構造をもつため、これは鎖状グラフィカルモデル(chain structure graphical model)とも呼ばれます。さらに、これは隠れマルコフモデル(hidden Markov model, HMM)とも呼ばれます^{†1}。

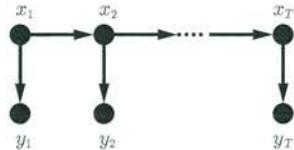


図 7.1 一般状態空間モデルのベイジアンネットワーク

一般状態空間モデルの特殊ケースとして、非線形・非ガウス型状態空間モデル(non-linear non-Gaussian state space model)があります^{†2}。非線形・非ガウス型状態空間モデルは、 $f_t(\cdot, \cdot)$, $h_t(\cdot, \cdot)$ を非線形関数、 w_t , v_t をそれぞれ(必ずしもガウス分布に従わない)白色雑音 W_t , V_t の実現値として、つぎで与えられます^{24), 86), 87)}。

定義 7.2 (非線形・非ガウス型状態空間モデル)

$x_t = f_t(x_{t-1}, w_t)$: システムモデル

$y_t = h_t(x_t, v_t)$: 観測モデル

^{†1} 隠れマルコフモデルでは状態 X_t が離散確率変数であることを想定するので、厳密には、隠れマルコフモデルは一般状態空間モデルの特別な場合といえます。

^{†2} 一般に、非線形モデルは線形モデルを含みますが、それと同様に、ここではガウス型を含んだモデルとして非ガウス型モデルを定義しています。

ここで、 W_t と V_t はそれぞれシステム雑音 (system noise)、観測雑音 (measurement noise) と呼ばれ、 W_t と V_t 、 W_t と X_s , $s < t$ および V_t と X_s , $s < t$ はいずれもたがいに独立であることを仮定します。

さらに特殊なケースとして、上記の非線形関数を線形関数に、白色雑音を白色ガウス雑音に限定することで、**線形・ガウス型状態空間モデル** (linear Gaussian state space model) が得られます。線形・ガウス型状態空間モデルは、 F_t , G_t , H_t を行列、 w_t , v_t をそれぞれ白色ガウス雑音 W_t , V_t の実現値として、つぎで与えられます^{24), 86), 87)}。

定義 7.3 (線形・ガウス型状態空間モデル)

$$\begin{aligned} x_t &= F_t x_{t-1} + G_t w_t && : \text{システムモデル} \\ y_t &= H_t x_t + v_t && : \text{観測モデル} \end{aligned}$$

ただし、 W_t と V_t , W_t と X_s , $s < t$ および V_t と X_s , $s < t$ はいずれも無相関であることを仮定します。

状態空間モデルは制御工学や電気回路の講義などで目にした方も多いと思いますが、多くの講義や教科書ではその背景についての説明なしに導入されるため、「なぜこのモデルを考えるのか?」、「なぜこのモデルがよいのか?」など疑問に感じたかもしれません。実は、状態空間モデルは

- 現在の状態は一つ前の時刻の状態にのみ依存する
- 現在の観測は現在の状態にのみ依存する

以上の二つの仮定だけから自然に導かれる確率モデルであるという点が重要です⁸⁸⁾。

7.2 予測分布、フィルタ分布、平滑化分布

状態推定では、状態の従う分布を逐次的に推定します。ここで、状態の従う分布

として図 7.2 に示されるように 3 種類の分布を考えます。予測分布 (predictive distribution) $p(x_t|y_{1:t-1})$ は時刻 $t - 1$ までの観測値 y が得られた下での状態 X_t の分布です。一方、フィルタ分布 (filtering distribution) $p(x_t|y_{1:t})$ は時刻 t までの観測値 y が得られた下での状態 X_t の分布です。また、平滑化分布 (smoothing distribution) $p(x_t|y_{1:T})$ はすべて時刻の観測値が得られた下での状態 X_t の分布で、次章の確率推論で説明する周辺事後分布に対応します。

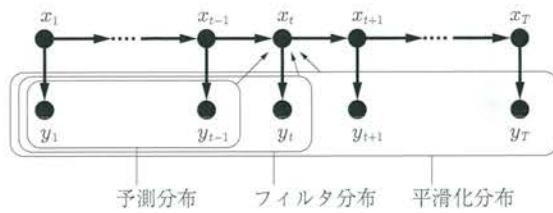


図 7.2 予測分布、フィルタ分布、平滑化分布

予測分布 $p(x_t|y_{1:t-1})$ は 1 時刻前のフィルタ分布 $p(x_{t-1}|y_{1:t-1})$ から求めることができます。また、フィルタ分布 $p(x_t|y_{1:t})$ は予測分布 $p(x_t|y_{1:t-1})$ と時刻 t での観測値 y_t から求めることができます。状態推定では観測のはじめから、予測分布とフィルタ分布の計算を交互に行っていきます。そして多くの場合、フィルタ分布を用いて状態を推定します。さらに高精度な状態推定値が必要な



図 7.3 時系列の状態推定の流れ

場合や状態推定に要する遅延が問題にならない場合は、観測の最後まで到達した後に平滑化という処理によって各時刻の平滑化分布を求めます。時刻 t の平滑化分布 $p(x_t|y_{1:T})$ は、時刻 $t+1$ の平滑化分布 $p(x_{t+1}|y_{1:T})$ および予測分布 $p(x_{t+1}|y_{1:t})$ と時刻 t のフィルタ分布 $p(x_t|y_{1:t})$ から計算することができます。状態推定におけるこれらの処理の流れを図 7.3 にまとめます。

以下で、予測分布、フィルタ分布、平滑化分布の具体的な計算方法について説明します。

7.2.1 予測分布

状態 X_{t-1} のフィルタ分布 $p(x_{t-1}|y_{1:t-1})$ から、状態 X_t の 1 ステップ予測分布 $p(x_t|y_{1:t-1})$ を求める問題を考えます。予測分布は周辺化演算によって

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int p(x_t, x_{t-1}|y_{1:t-1}) dx_{t-1} \\ &= \int p(x_t|x_{t-1}, y_{1:t-1}) p(x_{t-1}|y_{1:t-1}) dx_{t-1} \\ &= \int p(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1}) dx_{t-1} \end{aligned} \quad (7.5)$$

と書けます。ただし、2 行目から 3 行目への式変形において、マルコフ性（システムモデル） $p(x_t|x_{t-1}, y_{1:t-1}) = p(x_t|x_{t-1})$ を用いています。これより、状態 X_t の予測分布 $p(x_t|y_{1:t-1})$ が、状態 X_{t-1} のフィルタ分布 $p(x_{t-1}|y_{1:t-1})$ とシステムモデル $p(x_t|x_{t-1})$ によって表現されることがわかります。

7.2.2 フィルタ分布

つぎに、状態 X_t の 1 ステップ予測分布 $p(x_t|y_{1:t-1})$ と時刻 t における観測値 y_t から、状態 X_t のフィルタ分布 $p(x_t|y_{1:t})$ を求める問題を考えます。フィルタ分布は

$$p(x_t|y_{1:t}) = p(x_t|y_{1:t-1}, y_t) = \frac{p(x_t, y_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

$$\begin{aligned}
&= \frac{p(x_t, y_t | y_{1:t-1})}{\int p(x_t, y_t | y_{1:t-1}) dx_t} \\
&= \frac{p(y_t | x_t, y_{1:t-1}) p(x_t | y_{1:t-1})}{\int p(y_t | x_t, y_{1:t-1}) p(x_t | y_{1:t-1}) dx_t} \\
&= \frac{p(y_t | x_t) p(x_t | y_{1:t-1})}{\int p(y_t | x_t) p(x_t | y_{1:t-1}) dx_t} \tag{7.6}
\end{aligned}$$

と書けます。ただし、4行目から5行目への式変形において、式(7.3)のマルコフ性（観測モデル） $p(y_t | x_t, y_{1:t-1}) = p(y_t | x_t)$ を用いています。これより、時刻 t におけるフィルタ分布 $p(x_t | y_{1:t})$ が、時刻 $t-1$ における予測分布 $p(x_t | y_{1:t-1})$ と観測モデル $p(y_t | x_t)$ によって表現されることがわかります。

7.2.3 平滑化分布

最後に、状態 X_{t+1} の平滑化分布 $p(x_{t+1} | y_{1:T})$ から、状態 X_t の平滑化分布 $p(x_t | y_{1:T})$ を求める問題を考えます。平滑化分布は

$$\begin{aligned}
p(x_t | y_{1:T}) &= \int p(x_t, x_{t+1} | y_{1:T}) dx_{t+1} \\
&= \int p(x_t | x_{t+1}, y_{1:T}) p(x_{t+1} | y_{1:T}) dx_{t+1} \tag{7.7}
\end{aligned}$$

と書けます。ここで、 $p(x_t, y_{1:T} | x_{t+1})$ に対して確率の積法則を用いると

$$p(x_t, y_{1:T} | x_{t+1}) = p(x_t | x_{t+1}, y_{1:T}) p(y_{1:T} | x_{t+1}) \tag{7.8}$$

となります。一方、 $x_t, y_{1:t}$ と $y_{t+1:T}$ の x_{t+1} を与えた下での条件付き独立性を用いると

$$\begin{aligned}
p(x_t, y_{1:T} | x_{t+1}) &= p(x_t, y_{1:t}, y_{t+1:T} | x_{t+1}) \\
&= p(x_t, y_{1:t} | x_{t+1}) p(y_{t+1:T} | x_{t+1}) \\
&= p(x_t | x_{t+1}, y_{1:t}) p(y_{1:t} | x_{t+1}) p(y_{t+1:T} | x_{t+1}) \\
&= p(x_t | x_{t+1}, y_{1:t}) p(y_{1:t}, y_{t+1:T} | x_{t+1}) \\
&= p(x_t | x_{t+1}, y_{1:t}) p(y_{1:T} | x_{t+1}) \tag{7.9}
\end{aligned}$$

となります。したがって

$$p(x_t|x_{t+1}, y_{1:T}) = p(x_t|x_{t+1}, y_{1:t}) \quad (7.10)$$

が成り立つ (d-separation と呼ばれます) ので、これを用いると平滑化分布は

$$\begin{aligned} p(x_t|y_{1:T}) &= \int p(x_t|x_{t+1}, y_{1:t})p(x_{t+1}|y_{1:T})dx_{t+1} \\ &= \int \frac{p(x_t, x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T})dx_{t+1} \\ &= \int \frac{p(x_{t+1}|x_t, y_{1:t})p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T})dx_{t+1} \\ &= p(x_t|y_{1:t}) \int \frac{p(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T})dx_{t+1} \end{aligned} \quad (7.11)$$

と書けます。ただし、最後の行への式変形で、マルコフ性 (システムモデル)
 $p(x_{t+1}|x_t, y_{1:t}) = p(x_{t+1}|x_t)$ を用いています。

状態 X_t の平滑化分布 $p(x_t|y_{1:T})$ は、状態 X_{t+1} の平滑化分布 $p(x_{t+1}|y_{1:T})$
 と状態 X_t のフィルタ分布 $p(x_t|y_{1:t})$ 、状態 X_{t+1} の予測分布 $p(x_{t+1}|y_{1:t})$ 、シ
 ステムモデル $p(x_{t+1}|x_t)$ から計算されることがわかります。

以上 の方法によって、各時刻で予測分布やフィルタ分布、平滑化分布を求める際に周辺化が必要な確率変数の数が大幅に削減されるので、それぞれの分布をそれまでの計算結果を使用せずに毎回計算する素朴な方法に比べると計算量が大きく削減されます。しかしながら、ここで示した更新式には積分計算が含まれており、各分布が解析的に与えられない場合や計算量に強い制限がある場合には、そのままこれらの更新式を適用することは困難です。以下で述べる粒子フィルタでは確率分布を粒子で近似することで、またカルマンフィルタでは確率分布をガウス分布に限定することで、この問題を解決し現実的な状態推定法を実現しています。

7.3 粒子フィルタ

粒子フィルタ^{89), 90)} (particle filter) は、非線形・非ガウス型の状態空間モ

モデルにおいて予測分布とフィルタ分布の近似を逐次的に更新していくアルゴリズムです。重要な点は、確率密度関数を複数の「粒子（particle）」と呼ばれるサンプルを用いて近似することと、フィルタリングのステップにおいてサンプリング法の一つである **SIR** (sampling/importance resampling) のアイデア⁹⁾を利用することです。ここではその詳細について説明します。

粒子フィルタでは、一般に連続確率変数の条件付き確率密度関数である $p(x_t | y_{1:t-1})$ や $p(x_t | y_{1:t})$ を、粒子と呼ばれるそれぞれの分布からの実現値（サンプル）を用いて近似します。図 7.4 にその例を示します。図 (a) が元の確率密度関数を表し、図 (b) が図 (a) を近似する粒子を用いた確率質量関数を表します。図 (a) の確率密度関数の値が大きいところでは図 (b) の粒子が密に集まっていますが、図 (a) と図 (b) を比較すると、一見、粒子による表現が元の確率密度関数の近似になっているように見えません。しかし、図 (c) に示された図 (a) の確率密度関数に対応する累積分布関数と図 (d) に示された図 (b) の粒子による確率質量関数に対応する累積分布関数を比べると、粒子による表現は累積分布関数ではよい近似になっていることがわかります。

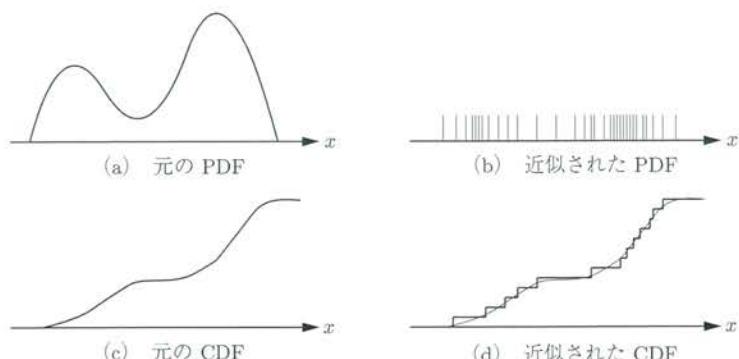


図 7.4 粒子による確率分布の近似

7.3.1 予測分布の計算

非線形・非ガウス型の状態空間モデル

$$x_t = f_t(x_{t-1}, w_t)$$

$$y_t = h_t(x_t, v_t)$$

を考えます。時刻 t の状態の予測分布 $p(x_t|y_{1:t-1})$ を近似する粒子を $\hat{x}_t^{(i)}$, $i = 1, 2, \dots, L$ とし、時刻 t の状態のフィルタ分布 $p(x_t|y_{1:t})$ を近似する粒子を $x_t^{(i)}$, $i = 1, 2, \dots, L$ とします。ここで L は粒子の数を表します。

式 (7.5) より、時刻 t の状態の予測分布は

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int \left\{ \int p(x_t, w_t|x_{t-1})dw_t \right\} p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int \left\{ \int p(x_t|x_{t-1}, w_t)p(w_t|x_{t-1})dw_t \right\} p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int \left\{ \int \delta(x_t - f_t(x_{t-1}, w_t)) p(w_t) dw_t \right\} p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ &\approx \frac{1}{L} \sum_{i=1}^L \delta(x_t - f_t(x_{t-1}^{(i)}, w_t^{(i)})) \end{aligned} \quad (7.12)$$

と書くことができます。ただし、 $\delta(\cdot)$ はデイラックのデルタ関数であり、 $w_t^{(i)}$ はシステム雑音の分布 $p(w_t)$ からの i 番目のサンプルを表します。式 (7.12) の 3 行目から 4 行目の変形では、非線形・非ガウス型の状態空間モデルでは一つ前の状態の実現値 x_{t-1} とシステム雑音の実現値 w_t が与えられると、つぎの時刻の状態 X_t が確定的に定まるという事実を利用しています。また、4 行目から 5 行目の変形では、 $p(w_t)$ と $p(x_{t-1}|y_{1:t-1})$ に従う有限の数の粒子（それぞれ L 個）を用いて積分計算を近似しています。

したがって、粒子フィルタにおける予測分布の計算は **Algorithm 7.1** で与えられます⁹⁾。

Algorithm 7.1 粒子フィルタによる予測分布の計算

- 時刻 $t-1$ の状態のフィルタ分布 $p(x_{t-1}|y_{1:t-1})$ を近似する粒子 $x_{t-1}^{(i)}$, $i = 1, 2, \dots, L$ を用意。
- システム雑音の分布 $p(w_t)$ からの L 個のサンプル $w_t^{(i)}$, $i = 1, 2, \dots, L$ を生成。
- 時刻 t の状態の予測分布 $p(x_t|y_{1:t-1})$ を近似する粒子をつぎで計算。

$$\hat{x}_t^{(i)} = f_t(x_{t-1}^{(i)}, w_t^{(i)}), \quad i = 1, 2, \dots, L \quad (7.13)$$

7.3.2 フィルタ分布の計算

つぎに、フィルタ分布の導出について考えます。式 (7.6) より、時刻 t の状態のフィルタ分布は

$$\begin{aligned} p(x_t|y_{1:t}) &= \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t} \\ &\propto p(y_t|x_t)p(x_t|y_{1:t-1}) \end{aligned} \quad (7.14)$$

と書くことができます。ここで、計算したいものは左辺の $p(x_t|y_{1:t})$ に従うサンプルですが、右辺の $p(x_t|y_{1:t-1})$ を近似するサンプルはすでにもっていることに注意します。すなわち、 $p(x_t|y_{1:t})$ がサンプルを生成したい所望の確率分布であり、 $p(x_t|y_{1:t-1})$ がサンプルを生成可能な分布（提案分布（proposal distribution）と呼ばれ、ここではすでにその分布からのサンプルをもっています）、観測モデルで与えられる尤度 $p(y_t|x_t)$ が所望の分布と提案分布の違いを表す重みであると考えると、7.3.3 項で説明する SIR と呼ばれるアルゴリズムをそのまま適用することで、 $p(x_t|y_{1:t-1})$ を近似するサンプルから $p(x_t|y_{1:t})$ を近似するサンプルを生成することができます。

具体的には、粒子フィルタにおけるフィルタ分布の計算は Algorithm 7.2 で与えられます⁹⁾。

Algorithm 7.2 粒子フィルタによるフィルタ分布の計算

- 時刻 t の状態の予測分布 $p(x_t|y_{1:t-1})$ を近似する粒子 $\hat{x}_t^{(i)}$, $i = 1, 2, \dots, L$ を用意。
- 確率質量が

$$\frac{p(y_t|\hat{x}_t^{(i)})}{\sum_{n=1}^L p(y_t|\hat{x}_t^{(n)})} \quad (7.15)$$

である、 $\hat{x}_t^{(i)}$ 上 ($i = 1, 2, \dots, L$) の離散分布から L 回サンプリングすることで時刻 t の状態のフィルタ分布 $p(x_t|y_{1:t})$ を近似する粒子 $x_t^{(i)}$, $i = 1, 2, \dots, L$ を得る。

重点サンプリング⁹⁾ (importance sampling) や SIR では提案分布と興味のある分布の形状が近い場合に計算効率がよいことが知られていますが、粒子フィ

ルタにおいては、それらはそれぞれ時刻 t の状態の予測分布とフィルタ分布なので、近い分布形状を有することが期待されます。

7.3.3 SIR (sampling/importance resampling)

粒子フィルタでは、SIR のアイデアに基づいて予測分布を近似するサンプルからフィルタ分布を近似するサンプルを生成するアルゴリズムが肝となっていました。ここでは、サンプルを容易に生成可能な提案分布 $q(x)$ があるときに、提案分布に従うサンプルから所望の分布 $p(x)$ に近似的に従うサンプルを生成するためのアルゴリズムである SIR について説明します。

なお、応用上、 $p(x)$ の値はある正規化定数 Z_p を除いた値だけがわかることが多いので、以下では

$$p(x) = \frac{1}{Z_p} \tilde{p}(x) \quad (7.16)$$

としたときに、任意の x に対して、 $\tilde{p}(x)$ の値はわかるが $p(x)$ の値はわからないものとして説明します。

SIR の手順を **Algorithm 7.3** に示します⁹⁾。

Algorithm 7.3 SIR アルゴリズム

- 容易にサンプルを抽出できる分布 $q(x)$ を用意（提案分布）。
 - 提案分布 $q(x)$ から L 個のサンプル $x^{(1)}, x^{(2)}, \dots, x^{(L)}$ を生成。
 - $w_l = \frac{\tilde{p}(x^{(l)}) / q(x^{(l)})}{\sum_{n=1}^L \tilde{p}(x^{(n)}) / q(x^{(n)})}$ によって重み w_1, w_2, \dots, w_L を決定。
 - 確率質量 w_1, w_2, \dots, w_L の離散分布 $(x^{(1)}, x^{(2)}, \dots, x^{(L)})$ に従う確率変数 X_r の L 個のサンプルを抽出する（リサンプリング）。
-

最後のステップ（リサンプリング（resampling））では、一様分布に従うサンプルを生成できれば、ここで所望の離散分布からのサンプルが生成できます。具体的には、 $\sum_{i=1}^L w_i = 1$ なので、区間 $[0 : 1]$ をそれぞれの区間の幅が w_1, w_2, \dots, w_L であるような L 個の区間に分割し、区間 $[0 : 1]$ の一様分布から生成したサンプルが l 番目の区間に含まれていれば $x^{(l)}$ を離散分布からのサンプルとして抽出します。ただし、SIR では L が有限の場合には近似的なサンプルでしかないことに注意します。

実際、リサンプリングされた確率変数 X_r の累積分布関数を計算すると

$$\begin{aligned}\Pr(X_r \leq a) &= \sum_{l: x^{(l)} \leq a} w_l \\ &= \frac{\sum_{l: x^{(l)} \leq a} \tilde{p}(x^{(l)})}{\sum_{n=1}^L \tilde{p}(x^{(n)})} \\ &= \frac{\frac{1}{L} \sum_{l=1}^L I(x^{(l)} \leq a) \tilde{p}(x^{(l)})}{\frac{1}{L} \sum_{n=1}^L \tilde{p}(x^{(n)})}\end{aligned}$$

となり、 $L \rightarrow \infty$ を考えると

$$\begin{aligned}\Pr(X_r \leq a) &\rightarrow \frac{\int I(x \leq a) \frac{\tilde{p}(x)}{q(x)} q(x) dx}{\int \frac{\tilde{p}(x)}{q(x)} q(x) dx} \\ &= \frac{\int I(x \leq a) \tilde{p}(x) dx}{\int \tilde{p}(x) dx} \\ &= \int I(x \leq a) p(x) dx\end{aligned}\tag{7.17}$$

となります。これは所望の分布 $p(x)$ の累積分布関数にほかなりません。

また、 $p(x)$ に従う確率変数 X に対して、その関数 $f(X)$ の期待値を求めたい場合には

$$\begin{aligned}E[f(X)] &= \int f(x) p(x) dx \\ &= \frac{\int f(x) \frac{\tilde{p}(x)}{q(x)} q(x) dx}{\int \frac{\tilde{p}(x)}{q(x)} q(x) dx}\end{aligned}$$

$$\approx \sum_{l=1}^L w_l f(x^{(l)}) \quad (7.18)$$

とすれば、 $q(x)$ に従うサンプル $x^{(l)}$ と重み w_l を用いて、関数 $f(X)$ の期待値の近似値が求まります。

一般に、 $q(x)$ の台が $p(x)$ の台を含んでいれば、 $q(x) \neq 0$ に対して

$$p(x) = \frac{p(x)}{q(x)} q(x)$$

と書けるので、SIRにおいてサンプルを生成したい所望の分布 $p(x)$ は提案分布 $q(x)$ に対して重み $p(x)/q(x)$ を乗算することで得られると解釈できます。

一方、粒子フィルタのフィルタ分布の計算では

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1})$$

なる関係があり、 $p(x_t|y_{1:t})$ がサンプルを生成したい所望の分布、 $p(x_t|y_{1:t-1})$ はすでにサンプルをもっている分布、 $p(y_t|x_t)$ が重みと考えることができます。つまり、SIRにおける重み $p(x)/q(x)$ を $p(y_t|x_t)$ に置き換え、さらに、提案分布 $q(x)$ から生成されるサンプルを、 $p(x_t|y_{1:t-1})$ を近似するサンプルで置き換えることで、 $p(x_t|y_{1:t})$ からのサンプルを生成する粒子フィルタのアルゴリズムが SIR によって構成されます。

7.4 カルマンフィルタ

カルマンフィルタ (Kalman filter) では、線形・ガウス型の状態空間モデル

$$x_t = F_t x_{t-1} + G_t w_t$$

$$y_t = H_t x_t + v_t$$

を考えます。ただし、システム雑音 W_t と観測雑音 V_t はいずれも白色ガウス雑音であり、 $E[W_t] = 0$, $E[W_t W_t^\top] = Q_t$, $E[V_t] = 0$, $E[V_t V_t^\top] = R_t$ とします。

初期状態 X_0 がガウス分布に従うと仮定すると、すべての状態 X_t および観測 Y_t はガウス分布に従います。また、予測分布 $p(x_t|y_{1:t-1})$ とフィルタ分布 $p(x_t|y_{1:t})$ のいずれもガウス分布となるため、線形・ガウス型の状態空間モデルでは状態 X_t の条件付き期待値と**条件付き共分散行列** (conditional covariance matrix) のみを追いかけていけばよいことになります。すなわち、条件付き期待値と条件付き共分散行列を

$$E[X_t|y_{1:s}] = \bar{x}_{t|s} \quad (7.19)$$

$$E[(X_t - \hat{x}_{t|s})(X_t - \bar{x}_{t|s})^\top | y_{1:s}] = P_{t|s} \quad (7.20)$$

と定義すると

$$p(x_t|y_{1:t-1}) = \mathcal{N}(\bar{x}_{t|t-1}, P_{t|t-1}) \quad (7.21)$$

$$p(x_t|y_{1:t}) = \mathcal{N}(\bar{x}_{t|t}, P_{t|t}) \quad (7.22)$$

であり、予測のステップでは $\bar{x}_{t-1|t-1}$, $P_{t-1|t-1}$ から $\bar{x}_{t|t-1}$, $P_{t|t-1}$ を、フィルタリングのステップでは $\bar{x}_{t|t-1}$, $P_{t|t-1}$ から $\bar{x}_{t|t}$, $P_{t|t}$ を、それぞれ更新します。

カルマンフィルタの具体的な更新の手続きは Algorithm 7.4 のとおりです²⁴⁾。

Algorithm 7.4 カルマンフィルタ

観測値 y_t を受けとったときのカルマンフィルタの処理は以下のとおりである。

- フィルタリング :

$$K_t = P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R_t)^{-1} \quad (7.23)$$

$$\bar{x}_{t|t} = \bar{x}_{t|t-1} + K_t(y_t - H_t \bar{x}_{t|t-1}) \quad (7.24)$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1} \quad (7.25)$$

- 予測 :

$$\bar{x}_{t+1|t} = F_t \bar{x}_{t|t} \quad (7.26)$$

$$P_{t+1|t} = F_t P_{t|t} F_t^\top + G_t Q_t G_t^\top \quad (7.27)$$

7.5 む す び

本章では、状態空間モデルに基づく手法について説明しました。状態空間モデルは通信分野の専門家にはあまり馴染みがないかもしれません、時系列解析の強力なツールで、時変チャネルの推定や位置推定、トラッキングなど通信に関連する問題においても幅広い応用が考えられます。次章で説明する確率伝播法は通信分野でよく利用されますが、本章で説明した粒子フィルタやカルマンフィルタは確率伝播法のスペシャルケースととらえられる手法ですので、併せて理解しておくとどこかで活用できる場面があるかもしれません。なお、本書ではカルマンフィルタのアルゴリズムの導出の詳細は説明しませんでしたが、標準的な導出法については文献24)などを参照してください。

8

確率推論

Next SIP

ベイズの定理は単なる条件付き確率の間の関係式にすぎませんが、そこに登場するそれぞれの確率変数に役割を与えることで、統計的信号処理や機械学習において中心的な役割を果たす重要な式になります。本章では、ベイズの定理に基づく確率推論の基礎事項について説明し、それを現実的な演算量で実現するためのアルゴリズムとしてさまざまな分野、場面で利用される確率伝播法について説明します。

8.1 確率推論問題

興味がある未知の離散確率変数を X 、観測可能な離散確率変数を Y とする
と、ベイズの定理

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \quad (8.1)$$

において左辺の $P(x|y)$ は事後確率分布、 $P(y|x)$ は尤度関数[†]、 $P(x)$ は事前確率分布と呼ばれます。ここで、事後確率分布 $P(x|y)$ は Y の実現値が観測されたという事象の下での未知の興味のある確率変数 X の確率分布を表しており、これを求めることが確率推論の主題となります。

例えば、確率変数 X が $x_j, j = 1, 2, \dots, N$ の値をとり得るとしたとき、すでに 4.6 節で見たように事後確率を最大化する x_j を推定値とする最大事後確率

[†] 確率変数 Y の実現値が与えられた下では、 $P(y|x)$ はもはや確率ではないため分布ではなく関数と呼ばれます。

推定

$$\hat{x}_{\text{map}} = \arg \max_{x_j} P(x_j|y) \quad (8.2)$$

は推定値が正しい確率を最大化、言い換えると、誤り確率を最小化します。また、事前確率 $P(x)$ が x_j によらず一定の場合は、尤度関数 $P(y|x)$ を最大化する x_j を推定値とする最尤推定

$$\hat{x}_{\text{ml}} = \arg \max_{x_j} P(y|x_j) \quad (8.3)$$

によっても誤り確率を最小化できます。

ピット誤り率の最小化や誤識別率の最小化は通信や機械学習において基本的であり、この例から事後確率の評価が重要であることがわかります。ただし、得られた事後確率分布の使用方法にはここで述べた最大事後確率推定以外にもさまざまなものがあり、問題や目的に応じて利用方法を選択する必要があることに注意します。

一般には観測される確率変数や未知の確率変数はいずれも複数存在します。このため、離散確率変数 X_1, X_2, \dots, X_n の同時確率分布を $P(x_1, x_2, \dots, x_n)$ とし、 X_1, X_2, \dots, X_n のうち $X_{m+1}, X_{m+2}, \dots, X_n$ の観測値が得られるとしたとき、最大事後確率推定の問題はつぎのように定義されます⁹⁾。

定義 8.1 (最大事後確率推定問題) 確率変数 X_1, X_2, \dots, X_n のうち $X_{m+1}, X_{m+2}, \dots, X_n$ の観測値（実現値） $a_{m+1}, a_{m+2}, \dots, a_n$ が与えられたとき、事後分布 $P(x_1, x_2, \dots, x_m | a_{m+1}, a_{m+2}, \dots, a_n)$ を最大にする x_1, x_2, \dots, x_m を求める問題、すなわち次式で表せる。

$$\begin{aligned} & \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m\} \\ &= \arg \max_{x_1, x_2, \dots, x_m} P(x_1, x_2, \dots, x_m | a_{m+1}, a_{m+2}, \dots, a_n) \quad (8.4) \end{aligned}$$

これは、ブロック誤り確率を最小にする推定法であり、**ブロックごと最大事後確率推定** (block-wise maximum a posteriori probability estimation) とも呼ばれます。

一方、周辺事後分布 (marginal posterior distribution) を考えることで、成
分毎の誤り確率を最小にする推定方法も考えられます。これは、**最大周辺事後確
率推定** (maximum posterior marginal estimation, **MPM 推定**)、あるいは成
分ごと最大事後確率推定 (symbol-wise maximum a posteriori probability
estimation) と呼ばれ、つぎで与えられます⁹¹⁾。

定義 8.2 (最大周辺事後確率推定) 確率変数 X_1, X_2, \dots, X_n のうち $X_{m+1}, X_{m+2}, \dots, X_n$ の観測値 (実現値) $a_{m+1}, a_{m+2}, \dots, a_n$ が与えられたとき、周辺事後分布 $P(x_i | a_{m+1}, a_{m+2}, \dots, a_n)$, $i = 1, 2, \dots, m$ を最大にする x_1, x_2, \dots, x_m を求める問題、すなわち次式で表せる。

$$\hat{x}_i = \arg \max_{x_i} P(x_i | a_{m+1}, a_{m+2}, \dots, a_n) \quad (8.5)$$

式 (8.4) のブロックごとの最大事後確率推定も式 (8.5) の成分ごとの最大周辺事後確率推定も、素朴に計算すると指数オーダーの計算量が必要となります。これは、式 (8.4) では、すべての x_1, x_2, \dots, x_m の組合せについて事後確率を評価する必要があり、各 X_i が q 通りの値をとり得る場合には、およそ q^{m-1} 回の演算が必要となることから明らかです。一方、式 (8.5) を直接計算して x_1 を評価しようとするとき

$$\begin{aligned} & \Pr(X_1 = x_1 | X_{m+1} = a_{m+1}, X_{m+2} = a_{m+2}, \dots, X_n = a_n) \\ &= \frac{\Pr(X_1 = x_1, X_{m+1} = a_{m+1}, X_{m+2} = a_{m+2}, \dots, X_n = a_n)}{\Pr(X_{m+1} = a_{m+1}, X_{m+2} = a_{m+2}, \dots, X_n = a_n)} \\ &= \alpha \sum_{x_2, x_3, \dots, x_m} p(x_1, x_2, \dots, x_m, a_{m+1}, a_{m+2}, \dots, a_n) \end{aligned} \quad (8.6)$$

となり (α は規格化定数)、やはり m について指数オーダーの計算量となりま

す。そのため、 m が大きい問題（例えば、誤り訂正符号の復号問題では数千にもなります）では、式(8.4)や式(8.5)を直接評価することが困難になります。しかし、式(8.5)については適切な確率モデルとその構造を利用すると周辺事後確率の計算を効率的に行うことが可能です。以下では、その基本的な手法である確率伝播法について説明します。

8.2 確率伝播法

確率伝播法 (belief propagation, BP) は確率伝搬法や信念伝播法とも呼ばれ、前章で考えた状態空間モデルよりも広いクラスの確率モデルにおける確率推論問題を解くことが可能な手法です。実際、状態空間モデルにおける平滑化分布は周辺事後分布そのものですので、前章で考えた状態推定法は平滑化まで含めると、ここで説明する確率伝播法の特別な場合（確率モデルが鎖状グラフィカルモデルの場合）に相当します。ここでは離散の確率変数を仮定して、分配法則 (distributive law) に基づく確率伝播法の計算原理について説明します。具体的な確率伝播法のアルゴリズムとして、ファクターグラフ上のメッセージ伝播アルゴリズムである sum-product アルゴリズムと、ペイジアンネットワーク上のアルゴリズムである Pearl の BP アルゴリズムの両方について述べ、それらの関係について簡単な例を用いて説明します。

8.2.1 確率伝播法の原理

離散確率変数 X_1, X_2, \dots, X_n の同時確率分布は、状態空間モデルの導出のときと同様に確率の積法則を繰り返し用いることで

$$\begin{aligned} P(x_1, x_2, \dots, x_n) &= P(x_1)P(x_2, x_3, \dots, x_n | x_1) \\ &= P(x_1)P(x_2 | x_1)P(x_3, x_4, \dots, x_n | x_1, x_2) \\ &\quad \vdots \\ &= P(x_1)P(x_2 | x_1)P(x_3 | x_1, x_2) \cdots \end{aligned}$$

$$\cdot P(x_n|x_1, x_2, \dots, x_{n-1}) \quad (8.7)$$

と書くことができます。

この変形だけでは周辺事後確率を求める際の計算量の削減にはなりませんが、多くの問題において考慮すべき確率変数の中に独立なものや条件付き独立なものも存在するため、式(8.7)の条件付き確率分布の条件に含まれる確率変数は必ずしもそのすべてが有効というわけではありません。この性質を利用することで周辺事後確率計算の演算量を削減することができます。

前章の状態空間モデルでは、その特別な場合として状態と観測のマルコフ性を利用しましたが、ここではより一般的の依存関係を仮定したときの確率推論について考えます。なお、状態空間モデルでは観測されない確率変数（状態）と観測される確率変数が一対一対応になっていたため、それらを区別して X と Y という記号をそれぞれの確率変数に使用していましたが、本章ではこのような対応関係は仮定しないため、確率変数の記号として X のみを使用し、それらのうちのいずれかが観測されるという状況を考えることにします。

簡単な例として五つの離散確率変数 X_1, X_2, \dots, X_5 の場合を考えます。同時分布は確率の積法則より

$$\begin{aligned} &P(x_1, x_2, x_3, x_4, x_5) \\ &= P(x_1)P(x_2|x_1)P(x_4|x_1, x_2)P(x_3|x_1, x_2, x_4)P(x_5|x_1, x_2, x_3, x_4) \end{aligned} \quad (8.8)$$

と書けます。ここで

$$X_2 \perp\!\!\!\perp X_4 | X_1 \quad (8.9)$$

$$X_3 \perp\!\!\!\perp X_1, X_2 | X_4 \quad (8.10)$$

$$X_5 \perp\!\!\!\perp X_1, X_2, X_3 | X_4 \quad (8.11)$$

なる条件付き独立性が存在するとすれば

$$P(x_4|x_1, x_2) = P(x_4|x_1) \quad (8.12)$$

$$P(x_3|x_1, x_2, x_4) = P(x_3|x_4) \quad (8.13)$$

$$P(x_5|x_1, x_2, x_3, x_4) = P(x_5|x_4) \quad (8.14)$$

となるので、同時分布は

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2|x_1)P(x_4|x_1)P(x_3|x_4)P(x_5|x_4) \quad (8.15)$$

となります。

この同時分布に対して、確率推論問題の具体例として $X_3 = a_3$ が観測されたときの確率変数 X_1 の周辺事後分布の計算を考えると

$$\begin{aligned} & \Pr(X_1 = x_1 | X_3 = a_3) \\ &= \frac{\Pr(X_1 = x_1, X_3 = a_3)}{\Pr(X_3 = a_3)} \\ &= \alpha \sum_{x_2, x_4, x_5} P(x_1, x_2, a_3, x_4, x_5) \\ &= \alpha \sum_{x_2, x_4, x_5} P(x_1)P(x_2|x_1)P(a_3|x_4)P(x_4|x_1)P(x_5|x_4) \\ &= \alpha P(x_1) \left(\sum_{x_2} P(x_2|x_1) \right) \left(\sum_{x_4} P(a_3|x_4) \left(\sum_{x_5} P(x_5|x_4) \right) \right) \end{aligned}$$

となります。ただし、 $\alpha = 1 / \Pr(X_3 = a_3)$ です。

周辺事後分布を同時分布から直接計算した場合には、各 X_i がとり得る値が q 通りあるとして $\mathcal{O}(q^3)$ の加算が必要であったのに対し、最後の式を利用すると $\mathcal{O}(q^2)$ にまで計算量が削減されていることがわかります。

計算量削減の肝は、分配法則を利用してグローバルな周辺化計算をローカルな周辺化計算にすることです。これは以下で説明する sum-product アルゴリズムや Pearl の BP アルゴリズムなどの確率伝播法の計算原理になっています。分配法則と確率伝播法についての詳しい議論は文献92) を参照してください。

8.2.2 sum-product アルゴリズム

sum-product アルゴリズムは木構造 (tree structure) をもつファクターグラフで表現された多変数関数の周辺化関数をファクターグラフ上で分散的に計算するメッセージ伝播アルゴリズムです。すでに見たように、同時確率分布を多変数関数とみなすことでそのファクターグラフ表現が得られますが、同様の考え方で sum-product アルゴリズムによって（事後）周辺分布を求めることができます。

n 変数の関数 f が

$$f(x_1, x_2, \dots, x_n) = f_1(A_1)f_2(A_2) \cdots f_m(A_m) \quad (8.16)$$

と m 個のローカル関数の積の形に因数分解されるとき、変数 x_r 以外のすべての変数について周辺化した関数

$$g_r(x_r) = \sum_{x_1, x_2, \dots, x_n \setminus x_r} f(x_1, x_2, \dots, x_n) \quad (8.17)$$

を求めることがあります。ただし、 A_1, A_2, \dots, A_m はそれぞれ $\{x_1, x_2, \dots, x_n\}$ の部分集合であり、 $x_1, x_2, \dots, x_n \setminus x_r$ は変数 x_r 以外について周辺化することを意味します。以下では、周辺化を考える多変数関数のファクターグラフは木構造である（サイクル（cycle）がない）ものとします。

sum-product アルゴリズムによる周辺化関数計算の手順は以下のとおりです⁹⁾。

1. $f(x_1, x_2, \dots, x_n)$ からファクターグラフを作成する。
2. 求めたい周辺化関数を $g_r(x_r)$ とするとき、変数ノード x_r を根とする木としてファクターグラフを描き直す。
3. 木の下側（葉側）のノードから上側（根側）のノードの順に、以下に説明する各ノードでの処理ルールに従って計算する。

ファクターグラフには変数ノードと関数ノードがありますが、各ノードでの処理ルールは異なり、それぞれ Algorithm 8.1 のとおりです。

行送りをこのページで
止めるため、表現を
このようにしましたが
よろしいですか？

Algorithm 8.1 sum-product アルゴリズム

変数ノードでの処理ルール：

変数ノード x_k からそれに直接つながった関数ノード f_i へ送るメッセージはつぎのように計算される（図 8.1）。

$$M_{x_k \rightarrow f_i}(x_k) = \prod_{a \in N(x_k) \setminus f_i} M_{a \rightarrow x_k}(x_k) \quad (8.18)$$

ただし、 $N(x_k)$ は変数ノード x_k に直接つながった関数ノードの集合であり、 $M_{a \rightarrow x_k}(x_k)$ は関数ノード a から変数ノード x_k に届いたメッセージを表す。 x_k が葉のときは

$$M_{x_k \rightarrow f_i}(x_k) = 1 \quad (8.19)$$

とする。

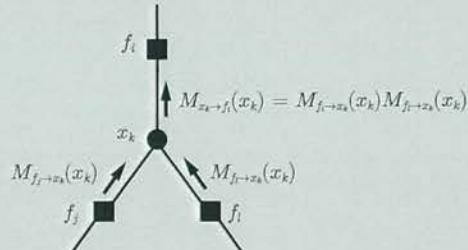


図 8.1 sum-product アルゴリズム（変数ノードでの処理）

関数ノードでの処理ルール：

関数ノード f_i からそれに直接つながった変数ノード x_k へのメッセージは

$$M_{f_i \rightarrow x_k}(x_k) = \sum_{A_i \setminus x_k} \left(f_i(A_i) \prod_{a \in A_i \setminus x_k} M_{a \rightarrow f_i}(a) \right) \quad (8.20)$$

と計算される（図 8.2）。 f_i が葉のときは

$$M_{f_i \rightarrow x_k}(x_k) = f_i(x_k) \quad (8.21)$$

とする。

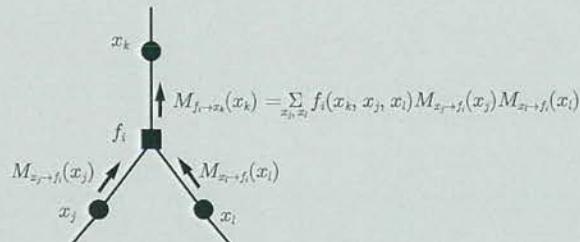


図 8.2 sum-product アルゴリズム（関数ノードでの処理）

変数ノード x_k から関数ノード f_i へ送られる $M_{x_k \rightarrow f_i}(x_k)$ や関数ノード f_i から変数ノード x_k へ送られる $M_{f_i \rightarrow x_k}(x_k)$ は x_k を引数にもつ関数なので、これらがグラフ上で送られるメッセージであるということの意味が理解しにくいかかもしれません。ここで考える周辺分布の計算では離散確率変数を考えているので、関数を送ることは具体的には確率変数がとり得る各離散値の確率（確率を成分にもつベクトル）の定数倍を送ることに対応します。なお、sum-product アルゴリズムの原理自体は連続確率変数に対しても有効ですが、グラフ上のメッセージ伝播の際に実際に通信路にメッセージを送信する必要がある場合などには、ガウス分布などのパラメトリックな関数を仮定するなどの工夫が必要になります。また、連続確率変数の場合には関数ノードでの周辺化計算（積分）を効率的に行うことも問題になることに注意します。

最後に根ノードでは **Algorithm 8.2** のようにして周辺化関数を得ます。

Algorithm 8.2 sum-product アルゴリズム

根ノードでの処理ルール：

最後に根ノードで

$$M_{x_r}(x_r) = \prod_{a \in N(x_r)} M_{a \rightarrow x_r}(x_r) = g_r(x_r) \quad (8.22)$$

と計算することで、所望の周辺化関数 $g_r(x_r)$ を得る。

ファクターグラフにサイクルが存在しない場合、上記の sum-product アルゴリズムによる分散的な処理によって所望の周辺化関数を厳密に求めることができます。以下ではその仕組みについて文献9)にならって説明します。

求めたい周辺化関数を

$$g_r(x_r) = \sum_{x_1, x_2, \dots, x_n \setminus x_r} f(x_1, x_2, \dots, x_n) \quad (8.23)$$

とします。さらに、変数ノード x_r （根）に直接つながっているある関数ノードを $a \in N(x_r)$ とし、関数ノード a を介して変数ノード x_r につながっているすべての変数ノードの集合を B_a とします（図 8.3）。

ファクターグラフの定義と、ここで考えるファクターグラフがサイクルをもたないことから、ある特定の変数ノード x_r に着目すると、多変数関数 $f(x_1, x_2, \dots, x_n)$

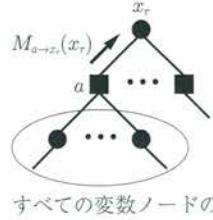


図 8.3 sum-product アルゴリズムの仕組み (1)

すべての変数ノードの集合 : B_a

は

$$f(x_1, x_2, \dots, x_n) = \prod_{a \in N(x_r)} F_a(x_r, B_a) \quad (8.24)$$

という因数分解の形で書けます。ここで、関数 $F_a(x_r, B_a)$, $a \in N(x_r)$ はそれぞれ変数ノード x_r に隣接する関数ノード a に対応する関数を因数としてもち、 x_r のみを共通の引数としてもつことに注意します。これを周辺化関数の式に代入すると

$$\begin{aligned} g_r(x_r) &= \sum_{x_1, x_2, \dots, x_n \setminus x_r} \prod_{a \in N(x_r)} F_a(x_r, B_a) \\ &= \prod_{a \in N(x_r)} \left(\sum_{B_a} F_a(x_r, B_a) \right) \\ &= \prod_{a \in N(x_r)} M_{a \rightarrow x_r}(x_r) \end{aligned} \quad (8.25)$$

を得ます。ただし

$$M_{a \rightarrow x_r}(x_r) \stackrel{\text{def}}{=} \sum_{B_a} F_a(x_r, B_a) \quad (8.26)$$

であり、 $M_{a \rightarrow x_r}(x_r)$ は関数ノード a 以下の情報のみから計算できます。つまり、 $M_{a \rightarrow x_r}(x_r)$ は関数ノード a から変数ノード x_r に送られるメッセージであると考えると、式 (8.25) は sum-product アルゴリズムの根ノードにおける処理を表していることがわかります。

式 (8.25) の 1 行目から 2 行目にかけての式変形は注意が必要です。一般に、和演算どうしはその順序を入れ替えることができますが、ここでは通常交換可

能でない和演算 Σ と積演算 Π の順序を入れ替わっています。これはファクターグラフがサイクルをもたないこと（言い換えると $\{x_r, B_a\}$, $a \in N(x_r)$ が x_r のみを共有すること）から可能な式変形で、sum-product アルゴリズムの根幹になっている部分ですので、簡単な例で確認してみましょう。

図 8.4 に示すような $N(x_r) = \{f_1, f_2\}$, $B_{f_1} = \{x_1, x_2\}$, $B_{f_2} = \{x_3, x_4\}$ なるファクターグラフを考えます。このとき

$$\sum_{x_1, x_2, x_3, x_4} \prod_{a \in \{f_1, f_2\}} F_a(x_r, B_a) \quad (8.27)$$

$$= \sum_{x_1, x_2, x_3, x_4} F_{f_1}(x_r, x_1, x_2) F_{f_2}(x_r, x_3, x_4) \quad (8.28)$$

$$= \sum_{x_1, x_2} \sum_{x_3, x_4} F_{f_1}(x_r, x_1, x_2) F_{f_2}(x_r, x_3, x_4) \quad (8.29)$$

となります。 $\{x_r, B_{f_1}\}, \{x_r, B_{f_2}\}$ が x_r のみを共有することから

$$\sum_{x_1, x_2, x_3, x_4} \prod_{a \in \{f_1, f_2\}} F_a(x_r, B_a) \quad (8.30)$$

$$= \left(\sum_{x_1, x_2} F_{f_1}(x_r, x_1, x_2) \right) \left(\sum_{x_3, x_4} F_{f_2}(x_r, x_3, x_4) \right) \quad (8.31)$$

$$= \prod_{a \in \{f_1, f_2\}} \sum_{B_a} F_a(x_r, B_a) \quad (8.32)$$

と変形できます。これが和と積の順序を入れ替えることが可能な原理です。

さて、sum-product アルゴリズムの仕組みを理解するための周辺化関数の変形の話に戻りましょう。図 8.4 のように、関数ノード a に直接接続する x_r 以外の変数ノードを $x_{a,1}, x_{a,2}, \dots, x_{a,L}$ とし、 $x_{a,l}$ を介して a につながっている

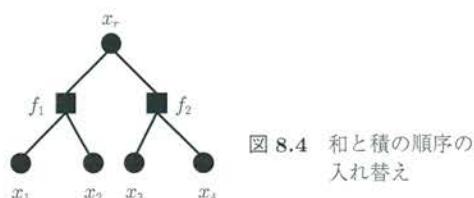


図 8.4 和と積の順序の入れ替え

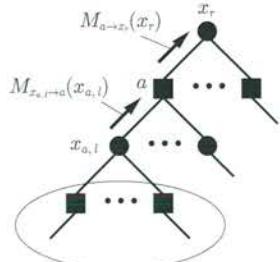


図 8.5 sum-product アルゴリズムの仕組み (2)

すべての変数ノードの集合を $B_{a,l}$ と定義すると

$$\begin{aligned}
 F_a(x_r, B_a) &= a(x_r, x_{a,1}, x_{a,2}, \dots, x_{a,L}) \\
 &\cdot G_{a,1}(x_{a,1}, B_{a,1})G_{a,2}(x_{a,2}, B_{a,2}) \cdots G_{a,L}(x_{a,L}, B_{a,L})
 \end{aligned} \tag{8.33}$$

と書くことができます。これより

$$\begin{aligned}
 M_{a \rightarrow x_r}(x_r) &= \sum_{B_a} a(x_r, x_{a,1}, x_{a,2}, \dots, x_{a,L}) \prod_{l=1}^L G_{a,l}(x_{a,l}, B_{a,l}) \\
 &= \sum_{x_{a,1}, x_{a,2}, \dots, x_{a,L}} a(x_r, x_{a,1}, x_{a,2}, \dots, x_{a,L}) \\
 &\quad \cdot \sum_{B_a \setminus \{x_{a,1}, x_{a,2}, \dots, x_{a,L}\}} \prod_{l=1}^L G_{a,l}(x_{a,l}, B_{a,l}) \\
 &= \sum_{x_{a,1}, x_{a,2}, \dots, x_{a,L}} a(x_r, x_{a,1}, x_{a,2}, \dots, x_{a,L}) \prod_{l=1}^L \sum_{B_{a,l}} G_{a,l}(x_{a,l}, B_{a,l}) \\
 &= \sum_{x_{a,1}, x_{a,2}, \dots, x_{a,L}} a(x_r, x_{a,1}, x_{a,2}, \dots, x_{a,L}) \prod_{x_{a,l} \in N(a) \setminus x_r} M_{x_{a,l} \rightarrow a}(x_{a,l})
 \end{aligned} \tag{8.34}$$

となります。ここでも、前述の和と積の順序の入れ替えを利用しました。また

$$M_{x_{a,l} \rightarrow a}(x_{a,l}) \stackrel{\text{def}}{=} \sum_{B_{a,l}} G_{a,l}(x_{a,l}, B_{a,l}) \quad (8.35)$$

であり、変数ノード $x_{a,l}$ 以下の情報のみから計算できる量なので、これを変数ノード $x_{a,l}$ から関数ノード a に送られるメッセージとすると、式(8.34)は sum-product アルゴリズムの関数ノード a での処理を表しています。

さらに、図 8.6 に示すように変数ノード $x_{a,l}$ に関数ノード b ($\neq a$) を介して接続している変数ノードの集合を $B_{a,l,b}$ とすると

$$G_{a,l}(x_{a,l}, B_{a,l}) = \prod_{b \in N(x_{a,l}) \setminus a} F_{a,b}(x_{a,l}, B_{a,l,b}) \quad (8.36)$$

と書けるので

$$\begin{aligned} M_{x_{a,l} \rightarrow a}(x_{a,l}) &= \sum_{B_{a,l}} \prod_{b \in N(x_{a,l}) \setminus a} F_{a,b}(x_{a,l}, B_{a,l,b}) \\ &= \prod_{b \in N(x_{a,l}) \setminus a} \sum_{B_{a,l,b}} F_{a,b}(x_{a,l}, B_{a,l,b}) \\ &= \prod_{b \in N(x_{a,l}) \setminus a} M_{b \rightarrow x_{a,l}}(x_{a,l}) \end{aligned} \quad (8.37)$$

となります。これは変数ノード $x_{a,l}$ での処理を表しています。

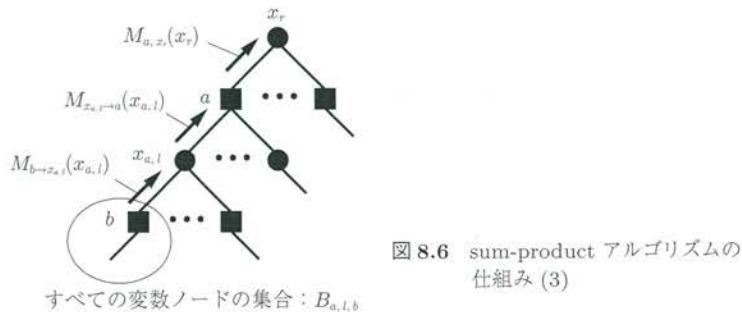


図 8.6 sum-product アルゴリズムの仕組み (3)

以上より、ファクターグラフにサイクルが存在しない場合には、sum-product アルゴリズムの変数ノードおよび関数ノードにおける分散的な処理ルールによって所望の周辺化関数を計算できることが示されました。

sum-product アルゴリズムでは、一般の多変数関数の周辺化関数を求める方

法を説明しましたが、確率推論において周辺事後確率を計算する際にはつぎのような手順になります。

1. 周辺事後分布 $P(x_r|x_1, x_2, \dots)$ を多変数関数 $f(x_r, x_1, x_2, \dots)$ とおく。
2. ファクターグラフを作成する。
3. sum-product アルゴリズムを実行する。

ただし、sum-product アルゴリズム中のメッセージは、例えば x_k の観測値が a_k であるときには $M_{x_k \rightarrow f_i}(x_k) = \delta_{x_k, a_k}$ (x_k が a_k の値をとる確率が 1 でそれ以外の値をとる確率は 0) とします。また、確率推論ではある特定の確率変数 x_r だけでなく、観測のないすべての確率変数についての周辺化関数を求める必要があるので、ファクターグラフの葉からある特定の根までメッセージを伝播させるのではなく、ファクターグラフのすべての葉から葉までメッセージを伝播させます。

また、上記の sum-product アルゴリズムではファクターグラフにサイクルがない場合について考えましたが、現実の確率推論の問題では多くの場合サイクルが存在します。サイクルがある場合に sum-product アルゴリズムでどのような解が得られるかについては理論的には完全にはわかっていませんが、大変興味深いことに、サイクルが存在する場合においても多くの応用例で良好な結果が得られることが経験的に知られています。なお、サイクルが存在する場合は、サイクルの影響で同じエッジに流れるメッセージが変化するので、ファクターグラフの端から端までのメッセージ伝播を複数回繰り返す必要があります。

サイクルが存在するときの確率伝播法（Loopy BP と呼ばれます）の収束特性の解析や理論的な理由づけの試みおよび対応策の検討としては、密度発展法⁹³⁾ (density evolution), ガウス近似法⁹⁴⁾ (Gaussian approximation), EXIT チャート法⁹⁵⁾ (extrinsic information transfer chart), 情報幾何 (information geometry) に基づく解釈^{96), 97)}, ベーテ自由エネルギー (Bethe free energy) と sum-product アルゴリズムの停留点の関連の指摘⁹⁸⁾, 一般化確率伝播法⁹⁹⁾ (generalized belief propagation) などが挙げられます。

8.2.3 Pearl の BP アルゴリズム

確率伝播法のアルゴリズムとして、8.2.2 項ではファクターグラフ上で定義される sum-product アルゴリズムを説明しました。これは sum-product アルゴリズムが簡潔で理解しやすく、周辺分布の計算だけでなく多変数関数の周辺化関数の計算というより一般的な目的に使用できるからです。また、sum-product アルゴリズムは、1960 年代初頭に Gallager によって低密度パリティ検査 (low density parity check, LDPC) 符号の復号アルゴリズム¹⁰⁰⁾ として最も初期に提案された確率伝播法であることもその理由の一つです。しかし、確率伝播法がより一般的な確率推論の枠組みで示されたのは、Pearl によって提案されたアルゴリズム¹⁰¹⁾ が最初とされています。そこで本項ではベイジアンネットワーク上のメッセージ伝播アルゴリズムである Pearl の BP アルゴリズムについて説明し、周辺分布の計算に sum-product アルゴリズムを適用したものと等価なアルゴリズムであることを簡単な例を用いて示します。

sum-product アルゴリズムは、変数ノードと関数ノードという 2 種類のノードをもつ無向グラフであるファクターグラフ上のアルゴリズムであり、ノードの種類で処理ルールが異なっていました。これに対して、Pearl の BP アルゴリズムは 1 種類のノードのみが存在する有向グラフであるベイジアンネットワーク上のアルゴリズムで、ノードの種類ではなく各ノードがメッセージを送る方

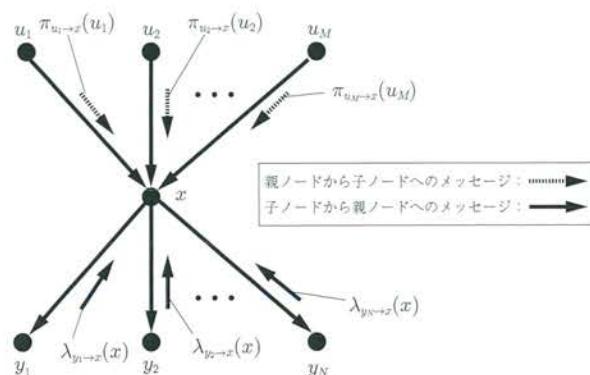


図 8.7 Pearl の BP アルゴリズム

向（親ノードに送るかそれとも子ノードに送るか）によってメッセージを処理するルールが異なっています。

図 8.7 のように複数の親ノード u_1, u_2, \dots, u_M と子ノード y_1, y_2, \dots, y_N をもつノード x における Pearl の BP アルゴリズムによる処理ルールを Algorithm 8.3 に示します。ただし、親ノード u_i から x に届くメッセージを $\pi_{u_i \rightarrow x}(u_i)$ とし、子ノード y_j から x に届くメッセージを $\lambda_{y_j \rightarrow x}(x)$ とします。

Algorithm 8.3 Pearl の BP アルゴリズム

ノード x が親ノード u_i に送るメッセージ：

$$\lambda_{x \rightarrow u_i}(u_i) = \sum_{u_1, u_2, \dots, u_M \setminus u_i} \left(\sum_x P(x|u_1, u_2, \dots, u_M) \cdot \prod_{k=1}^N \lambda_{y_k \rightarrow x}(x) \right) \cdot \prod_{k=1 \setminus i}^M \pi_{u_k \rightarrow x}(u_k) \quad (8.38)$$

ノード x が子 y_j に送るメッセージ：

$$\pi_{x \rightarrow y_j}(x) = \left(\prod_{k=1 \setminus j}^N \lambda_{y_k \rightarrow x}(x) \right) \cdot \sum_{u_1, u_2, \dots, u_M} P(x|u_1, u_2, \dots, u_M) \prod_{k=1}^M \pi_{u_k \rightarrow x}(u_k) \quad (8.39)$$

Pearl の BP アルゴリズムと sum-product アルゴリズムの関係を確認するために、図 8.8 に示す簡単なベイジアンネットワークの例でノード x から親ノード u_1 および子ノード y_1 に送られるメッセージ $\lambda_{x \rightarrow u_1}(u_1), \pi_{x \rightarrow y_1}(x)$ を Pearl

<

> ヤヤツ

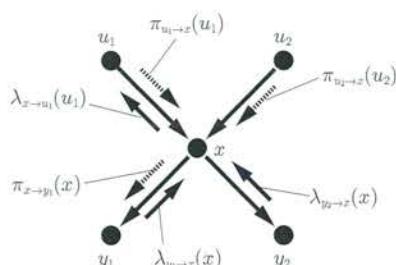


図 8.8 Pearl の BP アルゴリズムの例

のBPアルゴリズムの処理ルールに従って計算すると

$$\begin{aligned}\lambda_{x \rightarrow u_1}(u_1) &= \sum_{u_2} \left(\sum_x P(x|u_1, u_2) \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \right) \pi_{u_2 \rightarrow x}(u_2) \\ &= \sum_x \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \sum_{u_2} P(x|u_1, u_2) \pi_{u_2 \rightarrow x}(u_2)\end{aligned}\quad (8.40)$$

および

$$\pi_{x \rightarrow y_1}(x) = \lambda_{y_2 \rightarrow x}(x) \sum_{u_1, u_2} P(x|u_1, u_2) \pi_{u_1 \rightarrow x}(u_1) \pi_{u_2 \rightarrow x}(u_2)$$

となります。

一方、図8.8のペイジアンネットワークに対応する同時確率分布は

$$P(u_1, u_2, x, y_1, y_2) = P(x|u_1, u_2) P(y_1|x) P(y_2|x) P(u_1) P(u_2) \quad (8.41)$$

となるのでこれをファクターグラフ表現すると図8.9のようになります。

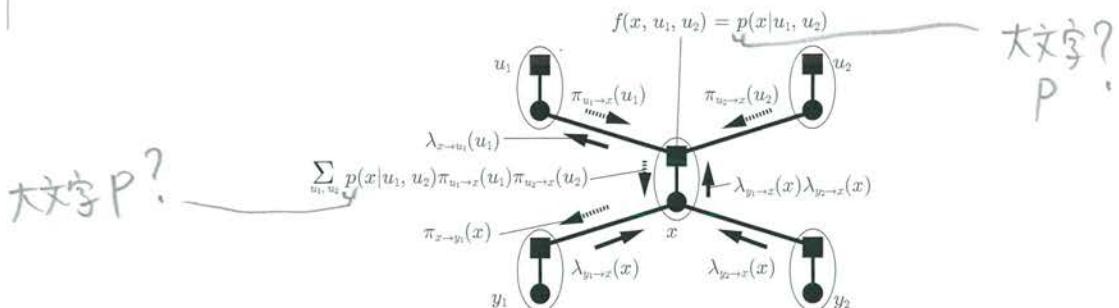


図 8.9 Pearl の BP アルゴリズムの例 (ファクターグラフ表現)

sum-productアルゴリズムのルールに従うと、変数ノード x から関数ノード $f(x, u_1, u_2)$ へのメッセージは、 $f(x, u_1, u_2)$ 以外の関数ノードからきたメッセージの積なので $\lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x)$ となります。一方、関数ノード $f(x, u_1, u_2)$ から変数ノード x へのメッセージは、 x 以外から届いたメッセージと自身の関数を乗算して、 x 以外の変数で周辺化したものですため

$$\sum_{u_1, u_2} P(x|u_1, u_2) \pi_{u_1 \rightarrow x}(u_1) \pi_{u_2 \rightarrow x}(u_2)$$

となります。よって図 8.9 のファクターグラフにおけるメッセージ $\lambda_{x \rightarrow u_1}(u_1)$ および $\pi_{x \rightarrow y_1}(x)$ はそれぞれ

$$\begin{aligned} \lambda_{x \rightarrow u_1}(u_1) &= \sum_{x, u_2} P(x|u_1, u_2) \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \pi_{u_2 \rightarrow x}(u_2) \\ &= \sum_x \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \sum_{u_2} P(x|u_1, u_2) \pi_{u_2 \rightarrow x}(u_2) \end{aligned} \quad (8.42)$$

$$\pi_{x \rightarrow y_1}(x) = \lambda_{y_2 \rightarrow x}(x) \sum_{u_1, u_2} P(x|u_1, u_2) \pi_{u_1 \rightarrow x}(u_1) \pi_{u_2 \rightarrow x}(u_2) \quad (8.43)$$

となり、Pearl の BP アルゴリズムによるメッセージと一致することが確認できます。

8.3 確率伝播法の応用

確率伝播法の応用例として、LDPC 符号とターボ符号の復号アルゴリズムと、高速フーリエ変換のアルゴリズムについて説明します。なお、8.3.1 項と 8.3.2 項ではガロア体（Galois field, GF）上のベクトルを扱うので、演算もガロア体上の演算であることに注意しましょう。

8.3.1 低密度パリティ検査 (LDPC) 符号

低密度パリティ検査 (low density parity check, LDPC) 符号はその復号アルゴリズムである sum-product アルゴリズムとともに 1960 年代初頭に Gallager によって提案された符号¹⁰⁰⁾です。LDPC 符号は、1990 年代に MacKay によって再発見¹⁰²⁾（この論文の中で、情報通信関連の論文で初めて “belief propagation” という言葉が使われました）されるまで長い間注目されませんでした。その理由の一つに、LDPC 符号が本領を發揮するのは符号長が十分に長い場合であり、そ

のような符号長の場合の特性を評価する計算機シミュレーションが当時の計算機では困難であったことが挙げられます。LDPC 符号は**符号長** (code length) や**符号化率** (code rate) によっては後述するターボ符号よりも特性がよいことが知られており、現在ではさまざまなシステムで採用されています。

線形符号の**符号語** (codeword) は、ガロア体上の長さ K の情報系列 (メッセージベクトル) を $\mathbf{u} = [u_1 \ u_2 \ \cdots \ u_K]^T \in \mathbb{F}_2$ としたとき

$$\mathbf{c} = [c_1 \ c_2 \ \cdots \ c_{K+N}]^T = \mathbf{G}\mathbf{u} \quad (8.44)$$

で生成されます。ここで、 $\mathbf{G} \in \mathbb{F}_2^{(K+N) \times K}$ は $(K+N) \times K$ の行列で、情報系列に N 個分の冗長成分を付加して符号語を生成するため、**生成行列** (generator matrix) と呼ばれます。また、 $K/(K+N)$ を符号化率といいます。これに対して、 \mathbf{G} の列ベクトルが張る空間の直交補空間を行ベクトルが張るように定義される $N \times (K+N)$ の行列を**パリティ検査行列** (parity check matrix) といいます。これは、受信語に誤り \mathbf{e} が存在すると

$$\mathbf{c}' = \mathbf{c} + \mathbf{e} \quad (8.45)$$

となります。これに検査行列 \mathbf{H} を乗算すると

$$\mathbf{H}\mathbf{c}' = \mathbf{H}\mathbf{c} + \mathbf{H}\mathbf{e} = \mathbf{H}\mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{e} = \mathbf{H}\mathbf{e} \quad (8.46)$$

となり、 $\mathbf{H}\mathbf{c}'$ から誤りの有無の検出や訂正ができるからです。ここで、検査行列と生成行列の間にある関係 $\mathbf{HG} = \mathbf{0}$ を用いています。

LDPC 符号は、非零要素の割合が小さいスパースな検査行列によって定義される線形符号です。例えば、検査行列が

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (8.47)$$

で与えられるるとすると、符号語 \mathbf{c} に対して一般に $\mathbf{H}\mathbf{c} = \mathbf{0}$ なる関係が成立することから、この符号は図 8.10 のようなグラフで表現されます[†]。これはタナーグラフ (Tanner graph) と呼ばれる 2 部グラフで、 \mathbf{H} の各列に対応する変数ノードと \mathbf{H} の各行に対応する検査ノードからなります。 \mathbf{H} の成分が 1 に対応するノードどうしがエッジで結ばれるため、タナーグラフの枝の数は \mathbf{H} の 1 の数に等しいという性質があります。

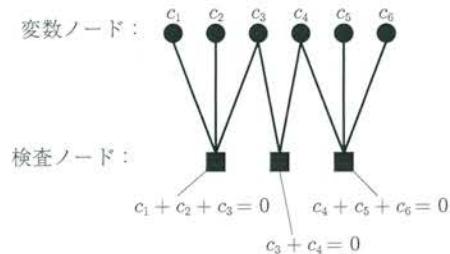


図 8.10 LDPC 符号の例（タナーグラフ表現）

受信語 c'_1, c'_2, \dots, c'_6 と送信符号語 c_1, c_2, \dots, c_6 を変数ノードとし、検査行列 \mathbf{H} の各行で与えられるパリティ検査 (\mathbf{H} の各行の 1 に対応する符号語をすべて足すと 0 になる性質) を閾数ノードとすると、この LDPC 符号のファクターグラフは図 8.11 で与えられます。このファクターグラフに、8.2.2 項で説明した sum-product アルゴリズムを適用することで、LDPC 符号の復号アルゴリズムが構成できます。ここで、sum-product アルゴリズムで正確な周辺事後分布が計算され、最尤復号が実現されるのは、ファクターグラフ（あるいはタナーグラフ）にサイクルが存在しない場合のみですが¹⁰³⁾、サイクルを含まない符号は適切に設計されたサイクルを含む符号に比べて最尤復号特性が劣ることが知られています¹⁰³⁾。一方、ファクターグラフに短いサイクルが存在する場合には、sum-product アルゴリズムによってよい復号特性が得られませんが、スパースな検査行列で構成される LDPC 符号の場合は、符号長が十分に長ければその

[†] この例では、検査行列の 1 の数と 0 の数にあまり違いがありませんが、実際の LDPC 符号は一般にもっとサイズが大きく、1 の割合が少ない検査行列が使用されます。

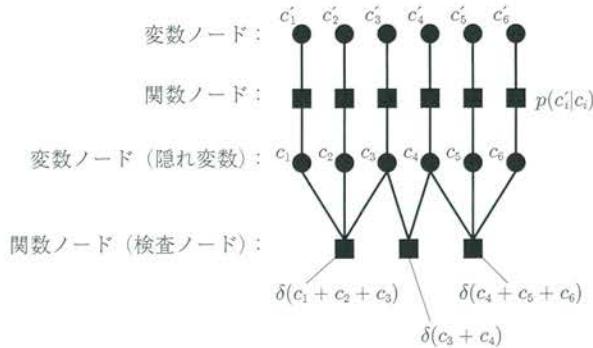


図 8.11 LDPC 符号の例 (ファクターグラフ表現)

ファクターグラフは短いサイクルをほとんど含まず、sum-product アルゴリズムによって良好な特性が得られることが経験的に知られています¹⁰²⁾。

8.3.2 ターボ符号

ターボ符号 (turbo code) とその復号アルゴリズムは、Berrou らによって 1993 年に提案され¹⁰⁴⁾、現実的な計算量でシャノン限界 (Shannon limit)[†] に迫る誤り率特性が得られることから大変注目を集めました。提案後、しばらくはターボ復号アルゴリズムの原理やそれによって良好な特性が得られる理由がわかつていませんでしたが、1990 年代後半に McEliece らによってターボ復号アルゴリズムは確率伝播法として解釈できることが示されました¹⁰⁵⁾。

ターボ符号は並列連接組織符号 (parallel concatenated systematic code) による符号化とターボ復号と呼ばれる二つの復号器による繰り返し復号処理を特徴とします。ここでは、組織符号 (systematic code) とその最大周辺事後確率復号について説明した後、並列連接組織符号、すなわちターボ符号とその最大周辺事後確率復号およびターボ復号アルゴリズムについて、文献¹⁰⁵⁾にならって Pearl の BP アルゴリズムを用いて説明します。

$U = (U_1, U_2, \dots, U_K)$ をシンボルからなる長さ K の情報系列 (を表す確率

[†] ある周波数帯域幅で雑音のある通信路において、誤りなく伝送可能な通信速度 (効率) の理論限界のこと、通信路容量 (channel capacity) とも呼ばれます。

変数) とし、これを符号化することで得られる符号語(を表す確率変数)を X とします。組織符号では情報系列がそのままの形で符号語に現れるため、その符号語は $X = (U, X_1)$ という形に書けます。以下では、 U と X_1 をそれぞれ符号語 X の組織部、非組織部と呼ぶことにします。

符号語 X が通信路を通って受信された受信語を $Y = (Y_s, Y_1)$ とします(図 8.12)。ここで、 Y_s および Y_1 はそれぞれ、 X の組織部および非組織部に対応する受信語です。

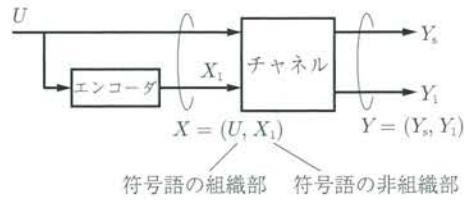


図 8.12 組織符号

通信路を表す条件付き確率分布を $P(y|x) \stackrel{\text{def}}{=} \Pr(Y = y|X = x)$ と定義し、通信路を無記憶(memoryless)[†]と仮定すると

$$\begin{aligned} P(y|x) &= P(y_s, y_1|u, x_1) \\ &= P(y_s|u)P(y_1|x_1) \\ &= \left(\prod_{i=1}^K P(y_{si}|u_i) \right) \cdot P(y_1|x_1) \end{aligned} \quad (8.48)$$

と書くことができます。ただし、 y_{si} は y_s の i 番目の成分を表します。

Y_s, Y_1 の実現値(受信信号) y_s, y_1 を得たときのシンボルごと最大事後確率復号(最大周辺事後確率復号)は周辺事後確率

$$\text{BEL}_i(a) \stackrel{\text{def}}{=} \Pr\{U_i = a|Y_s = y_s, Y_1 = y_1\} \quad (8.49)$$

[†] 送受信信号をそれぞれ $(x_1, x_2, \dots, x_n), (y_1, y_2, \dots, y_n)$ としたとき、 $i = 1, 2, \dots, n$ に対して x_i が与えられたときに y_i が $(x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ および $(y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_n)$ と独立である場合に、この通信路は無記憶であるといいます。

に基づいて行われます。ここで、 $a \in A$ であり、 A は情報シンボルのアルファベットです。尤度関数と事前確率をそれぞれ

$$\lambda_{y_{si} \rightarrow u_i}(u_i) = P(y_{si}|u_i) \quad (8.50)$$

$$\pi_{u_i}(a) = \Pr(U_i = a) \quad (8.51)$$

と定義し、式 (8.48) を用いると、周辺事後確率 $\text{BEL}_i(a)$ は

$$\begin{aligned} \text{BEL}_i(a) &= \alpha \sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1) \prod_{j=1}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \\ &= \alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}(a) \sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1) \\ &\quad \cdot \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \end{aligned} \quad (8.52)$$

と書くことができます。ただし、 α は Pearl の α 表記と呼ばれ¹⁰¹⁾、 $\text{BEL}_i(a)$ が確率分布（合計が 1）になるようにするための規格化定数です。

式 (8.52) の $\lambda_{y_{si} \rightarrow u_i}(a)$ は符号語の組織部分の受信信号から得られる尤度です。また、 $\pi_{u_i}(a)$ は事前確率で、残りの部分、すなわち

$$\sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)$$

は外部値 (extrinsic value) と呼ばれます。後に説明するターボ符号では外部値がきわめて重要な役割を果たします。

周辺事後確率の式 (8.52) から組織符号の復号問題をペイジアンネットワークで表現すると図 8.13 のようになります。このネットワークに Pearl の BP アルゴリズムを適用することを考えてみましょう。組織部分の受信信号に対応するノード y_{si} からは $\lambda_{y_{si} \rightarrow u_i}(u_i) = P(y_{si}|u_i)$ が情報系列に対応するノード u_i にメッセージとして送られ、同様に非組織部の受信信号 y_1 からは符号語の非組織部のノード x_1 に $P(y_1|x_1)$ がメッセージとして送られます。 u_i は子ノード

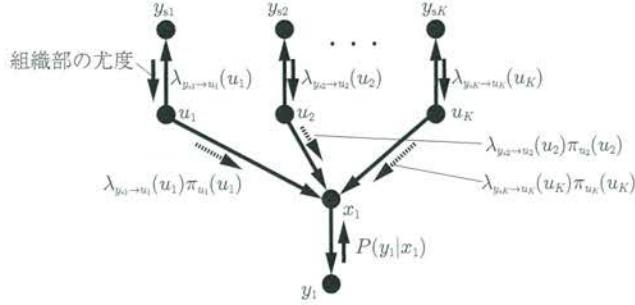


図 8.13 組織符号のベイジアンネットワーク表現

ドのみがある（親ノードをもたない）ノードなので、 y_{si} から届いたメッセージ $\lambda_{y_{si} \rightarrow u_i}(u_i)$ に自身がもつ事前確率 $\pi_{u_i}(u_i)$ を乗算したものをメッセージとして x_1 に送ります。つぎに、すべての u_i からメッセージを受けとった x_1 が各 u_i に送るメッセージは

$$\begin{aligned} & \sum_{u_1, u_2, \dots, u_K : u_i = a} (P(x_1|u)P(y_1|x_1)) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \\ &= \sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \end{aligned} \quad (8.53)$$

となります。ここで、 u と x_1 の関係は確定的であることに注意します。

最後にノード u_i では、届いたすべてのメッセージと自身のもつ事前確率を乗算し正規化することで

$$\alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}(a) \sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \quad (8.54)$$

を得ます。これは式 (8.52) の所望の周辺事後確率に一致していることがわかります。

つぎに、図 8.14 に示す並列接続組織符号（ターボ符号）とその最大周辺事後確率復号について考えてみましょう。ここで、 X_1, X_2 は二つの符号器の出

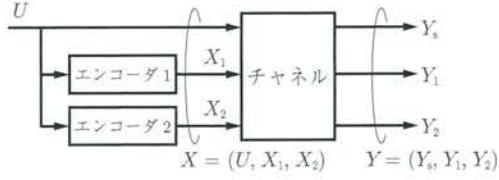


図 8.14 並列連接組織符号（ターボ符号）

力を表す確率変数で、符号語 $X = (U, X_1, X_2)$ の非組織部です。また、 Y_1, Y_2 は X_1, X_2 に対応する受信語です。無記憶通信路を仮定することで、通信路を表す条件付き確率は

$$\begin{aligned}
 P(y|x) &= P(y_s, y_1, y_2|u, x_1, x_2) \\
 &= P(y_s|u)P(y_1|x_1)P(y_2|x_2) \\
 &= \left(\prod_{i=1}^K P(y_{si}|u_i) \right) P(y_1|x_1)P(y_2|x_2)
 \end{aligned} \tag{8.55}$$

と書くことができます。さらに、組織符号のときと同様に尤度関数 $\lambda_{y_{si} \rightarrow u_i}(u_i)$ と事前確率 $\pi_{u_i}(u_i)$ を定義すると、シンボルごと最大事後確率復号（最大周辺事後確率復号）を実現するための周辺事後確率は

$$\begin{aligned}
 \text{BEL}_i(a) &= \alpha \sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1)P(y_2|x_2) \prod_{j=1}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \\
 &= \alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}(a) \sum_{u_1, u_2, \dots, u_K : u_i = a} P(y_1|x_1)P(y_2|x_2) \\
 &\quad \cdot \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)
 \end{aligned} \tag{8.56}$$

と書くことができます。

ターボ復号では周辺事後分布（の近似）を得るために、符号語 (U, X_1) および (U, X_2) に対応するシンボルごと最大事後確率復号を交互に何度も行い、復号結果（正確には外部値）をつぎのステップでの事前確率に取り込むという形でたがいの情報を交換するアルゴリズムを用います。具体的には、奇数番目のス

テップすなわち $2m - 1$ 番目のステップでは、事前確率を $P(u_i) = \pi_{u_i}^{(2m-2)}(u_i)$ として以下の周辺事後確率に基づく符号語 (U, X_1) の最大周辺事後確率復号を行います。

$$\begin{aligned} & \Pr\{U_i = a | Y_s = y_s, Y_1 = y_1\} \\ &= \alpha \sum_{u_1, u_2, \dots, u_K: u_i = a} P(y_1 | x_1) \prod_{j=1}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(2m-2)}(u_j) \\ &= \alpha \lambda_i(a) \pi_{u_i}^{(2m-2)}(a) \sum_{u_1, u_2, \dots, u_K: u_i = a} P(y_1 | x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(2m-2)}(u_j) \end{aligned} \quad (8.57)$$

ここで $\pi_{u_i}^{(2m-2)}(u_i)$ は $2m - 2$ 番目の (U, X_2) の復号結果から得られた外部値で、繰り返し処理の初回の $m = 1$ の場合には通常一様分布が用いられます。

つぎに、式 (8.57) の外部値 (式 (8.57) 右辺の $\alpha \lambda_i(a) \pi_{u_i}^{(2m-2)}(a)$ 以外の部分) を $\pi_{u_i}^{(2m-1)}(u_i)$ としてつぎのステップに受け渡します。さらに、 $2m$ 番目のステップでは $P(u_i) = \pi_{u_i}^{(2m-1)}(u_i)$ として (U, X_2) の周辺事後確率

$$\begin{aligned} & \Pr(U_i = a | Y_s = y_s, Y_2 = y_2) \\ &= \alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}^{(2m-1)}(a) \sum_{u_1, u_2, \dots, u_K: u_i = a} P(y_2 | x_2) \\ & \cdot \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(2m-1)}(u_j) \end{aligned} \quad (8.58)$$

を計算し、その外部値をつぎのステップでの事前確率 $P(u_i) = \pi_{u_i}^{(2m)}(u_i)$ として受け渡します。以上の処理をある終了条件を満たすまで繰り返し、最後に得られている周辺事後確率から復号結果を出力します。

以上のように、ターボ復号は符号語 (U, X_1) および (U, X_2) の復号を交互に行うことによって外部値を更新していくアルゴリズムですが、各回における復号結果、すなわち周辺事後確率の推定値、ではなくその一部分である外部値をつぎのス

テップの事前確率として利用する理由が明らかではないように思えます。そこで、ターボ符号の周辺事後分布を図 8.15 のようにベイジアンネットワークで表現し、このベイジアンネットワークに Pearl の BP アルゴリズムを適用することを考えてみます。

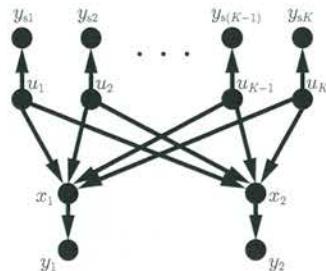


図 8.15 ターボ符号のベイジアン
ネットワーク表現

まず、先ほどの組織符号の場合と異なり、ベイジアンネットワークにサイクルが存在するため、BP アルゴリズムによって真の周辺事後確率が得られないことに注意しましょう。また、BP アルゴリズムのオリジナルの手順では、メッセージを送信するエッジ以外のすべてのエッジからのメッセージが届いて初めてメッセージの計算および送信ができますが、サイクルが存在する場合には、そのようなルールではアルゴリズムが途中で止まってしまう箇所が発生するため、メッセージの人為的な初期化が必要となります。さらに、サイクルが存在する場合にはメッセージを処理、伝播する順序によって異なる結果が得られるため、BP アルゴリズムを適用するためにはそのスケジューリングを決定する必要があります。そこで、ターボ符号の復号は、まずつぎのような手順で行われます。

- ノード x_1 からノード u_i に届くメッセージを $\lambda_{x_1 \rightarrow u_i}(u_i) = 1$ 、ノード x_2 からノード u_i に届くメッセージを $\lambda_{x_2 \rightarrow u_i}(u_i) = 1$ とする。
- ノード y_{si} および y_1, y_2 は葉のノードなので繰り返し処理に関係なくつねに固定のメッセージ $\lambda_{y_{si} \rightarrow u_i}(u_i), P(y_1|x_1)$ および $P(y_2|x_2)$ をそれぞれ送る。
- ノード u_i がもつ事前確率 $\pi_{u_i}(u_i)$ も固定で一様分布とする。

- まず、ノード u_1, u_2, \dots, u_K でメッセージを計算し、 x_1 および x_2 に送信する（ノード u のアクティベーションという）。ここで、 u_i から x_1 および x_2 へのメッセージはいずれも $\alpha \lambda_{y_{si} \rightarrow u_i}(u_i)$ である。
- この時点ではノード x_1 と x_2 はすべてのエッジからメッセージが届いているのでどちらもメッセージの更新が可能であるが、まず x_1 のアクティベーションを行うことになると、 x_1 から u_i へのメッセージは

$$\begin{aligned} & \alpha \sum_{u_1, u_2, \dots, u_K: u_i} \left(\sum_{x_1} P(x_1|u) P(y_1|x_1) \right) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \\ & = \alpha \sum_{u_1, u_2, \dots, u_K: u_i} P(y_1|x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \end{aligned} \quad (8.59)$$

と計算される。これを $\pi_{u_i}^{(1)}$ とする（これまで π は外部値の表記に用いてきたが、実際、これはターボ復号アルゴリズムの外部値 $\pi_{u_i}^{(1)}$ に一致している）。

- これより、ノード u に届くメッセージの一部が更新されたので、つぎは u のアクティベーションを行う。 x_1 から届くメッセージのみが更新されているので、各ノード u_i が更新すべきメッセージは x_2 宛のものだけであり、 $\alpha \lambda_{y_{si} \rightarrow u_i}(u_i) \pi_{u_i}^{(1)}$ となる。
- 続いて、 x_2 のアクティベーションを行うと、 x_2 から u_i へのメッセージは

$$\begin{aligned} & \alpha \sum_{u_1, u_2, \dots, u_K: u_i} \left(\sum_{x_2} P(x_2|u) P(y_2|x_2) \right) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(1)} \\ & = \alpha \sum_{u_1, u_2, \dots, u_K: u_i} P(y_2|x_2) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(1)} \end{aligned} \quad (8.60)$$

と計算され、これは外部値 $\pi_{u_i}^{(2)}$ に一致していることがわかる。

以下同様に、ノードのアクティベーションを $u \rightarrow x_1 \rightarrow u \rightarrow x_2 \rightarrow \dots$ の順に繰り返していくと、上記の初期化および順序による Pearl の BP アルゴリズムはターボ復号のアルゴリズムが構成されます。

さて、改めて外部値を次段の事前値とする理由について考えてみましょう。ターボ復号では要素符号 (U, X_1) および (U, X_2) のそれぞれの復号を交互に行うため、例えば要素符号 (U, X_1) の復号アルゴリズムからはペイジアンネットワークが図 8.15 ではなく図 8.16 のように見えています。このときどのようにすればもう一つの符号語のもつ情報をうまく取り込むことができるでしょうか。

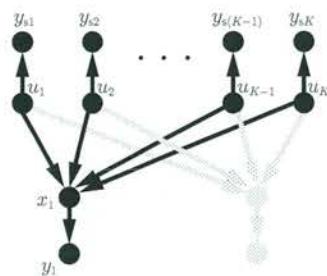


図 8.16 要素符号 (U, X_1) の復号アルゴリズムから見えるターボ符号のペイジアンネットワーク表現（ノード x_2 および y_2 に対応する部分が見えていない）

図 8.17 にそのアイデアを説明するためのペイジアンネットワークを示します。図 (a) はターボ符号の元のペイジアンネットワークの u_i にかかる箇所を抽出したものです。一方、ターボ符号の復号アルゴリズムにおける要素符号 (U, X_1) の復号の際には x_2 は見えておらず、ネットワークは図 (b) のようになっています。ここで、両方のネットワークでノード x_1 に届けられるメッセージが同一になるようにすることを考えます。 u_i は子ノードのみをもつノードであることから、 x_1 に送るメッセージはそれ以外のノードからのメッセージと事前確率 $P(u_i)$ の積となるため、右の x_2 のノードが省略されたネットワークでは x_2 から届くはずであったメッセージ $\pi_{u_i}^{(2m-2)}$ を事前確率 $p(u_i)$ に含んでし

図(b)

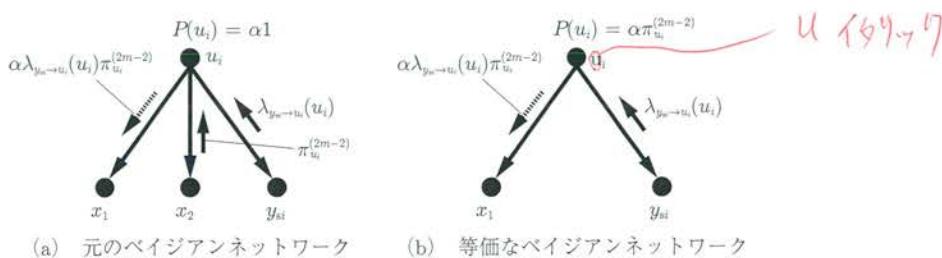


図 8.17 外部値を次段の事前値とする理由

まえばよいことがわかります。この処理は外部値をつぎのステップの事前確率にしていることにはなりません。このようにターボ符号およびその復号アルゴリズムは Pearl の BP アルゴリズムを用いて解釈することが可能で、これによりその仕組みを見通しよく理解することができます。

8.3.3 高速フーリエ変換 (FFT)

最後に、確率伝播法の特別な場合と解釈することができる例として、高速フーリエ変換 (fast Fourier transform, FFT) のアルゴリズムを取り上げます。

FFT は離散フーリエ変換

$$W_k = \sum_{n=0}^{N-1} w(n)e^{-j\frac{2\pi}{N}nk} \quad (8.61)$$

を効率的に演算するためのアルゴリズムです。DFT を計算するために式 (8.61) を直接的に計算すると N^2 のオーダーの計算量が必要となります

$$\begin{aligned} W_k &= \sum_{n=0}^{\frac{N}{2}-1} w(2n)e^{-j\frac{2\pi}{N}2nk} + \sum_{n=0}^{\frac{N}{2}-1} w(2n+1)e^{-j\frac{2\pi}{N}(2n+1)k} \\ &= \sum_{n=0}^{\frac{N}{2}-1} w(2n)e^{-j\frac{2\pi}{N}nk} + e^{-j\frac{2\pi}{N}k} \sum_{n=0}^{\frac{N}{2}-1} w(2n+1)e^{-j\frac{2\pi}{N}nk} \quad (8.62) \end{aligned}$$

と変形することで N 点の DFT 計算を二つの $N/2$ 点 DFT 計算で表現し、さらにそれぞれの $N/2$ 点 DFT の離散周波数領域における周期性（周期 $N/2$ ）を利用して、結果的に演算量が $(N/2)^2 \times 2 = N^2/2$ のオーダーに削減できるというのが FFT の基本的なアイデアです。 N が 2 の整数べき乗のとき、上記の分割処理を 2 点 DFT になるまで繰り返していくことで、最終的に $N \log N$ の演算量にまで削減することができます。

Cooley と Tukey の 1960 年代の発明¹⁰⁶⁾ とされている†この有名な FFT アルゴリズムもまたファクターグラフを用いた確率伝播法として解釈することができます。

† Gauss が 1800 年代に FFT と同様のアルゴリズムを発見していたという指摘もあります¹⁰⁷⁾。

できます^{92), 108)}。FFT アルゴリズムのファクターグラフ表現をするために、8 点 DFT の場合を考えてみます。

離散時間系列 $w(n)$ の 8 点 DFT は、 $\phi_N = 2\pi/N$ とし、 n および k を 2 進数表現した $n = 4x_2 + 2x_1 + x_0$, $k = 4y_2 + 2y_1 + y_0$ を用いることで

$$\begin{aligned}
 W_k &= \sum_{n=0}^7 w(n) e^{-j\phi_s n k} \\
 &= \sum_{x_0, x_1, x_2} w(4x_2 + 2x_1 + x_0) e^{-j\phi_s (4x_2 + 2x_1 + x_0)(4y_2 + 2y_1 + y_0)} \\
 &= \sum_{x_0, x_1, x_2} w(4x_2 + 2x_1 + x_0) \\
 &\quad \cdot (-1)^{x_2 y_0} (-1)^{x_1 y_1} (-1)^{x_0 y_2} (j)^{-x_0 y_1} (j)^{-x_1 y_0} e^{-j\phi_s (x_0 y_0)} \\
 &= \sum_{x_0} (-1)^{x_0 y_2} (j)^{-x_0 y_1} e^{-j\phi_s (x_0 y_0)} \sum_{x_1} (-1)^{x_1 y_1} (j)^{-x_1 y_0} \\
 &\quad \cdot \sum_{x_2} w(4x_2 + 2x_1 + x_0) (-1)^{x_2 y_0} \tag{8.63}
 \end{aligned}$$

と変形することができます。

この多変数関数をそのままファクターグラフ表現すると図 8.18 のようになりますが⁶、これにはサイクルが含まれているため sum-product アルゴリズムをそのまま適用しても厳密な周辺化関数を求めることができません。そこで、図 8.18 からスパンニング木 (spanning tree) を構成し

$$a(x_2, y_0) = (-1)^{x_2 y_0}$$

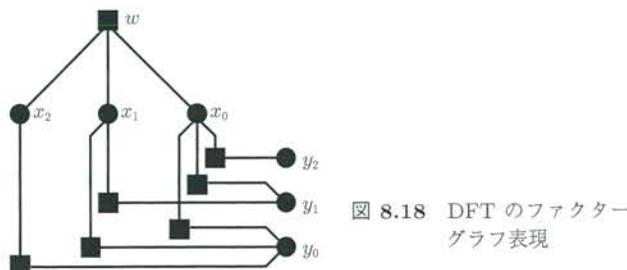


図 8.18 DFT のファクターグラフ表現

$$b(x_1, y_0, y_1) = (-1)^{x_1 y_1} (-j)^{x_1 y_0}$$

$$c(x_0, y_0, y_1, y_2) = (-1)^{x_0 y_2} (-j)^{x_0 y_1} e^{j\phi_s(x_0 y_0)}$$

と関数を定義（クラスタリング（clustering））することで、図 8.19 に示される修正版のファクターグラフを得ることができます（ファクターグラフのスパニング木の詳細については文献108）などを参照してください）。

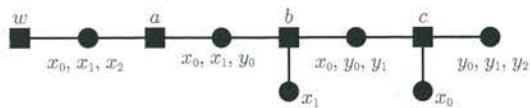


図 8.19 DFT のファクターグラフ表現（修正版）

変数ノード中の x_0, x_1 は冗長な表記となっていますが、元のグラフ中でのつながりを明示するために追加しています。実は、このファクターグラフに sum-product アルゴリズムを適用すると、FFT と同一のアルゴリズムが得られることが確認できます。実際、図 8.19 の左側から計算を進めると、 x_2, x_1, x_0 の順に 3 段階で周辺化が行われ、 $w(n)$ から W_k が得られます。

8.4 むすび

本章では、確率推論問題を導入し、最大周辺事後確率推定を行うための確率伝播法について説明しました。確率伝播法は誤り訂正符号の復号アルゴリズム¹⁰⁰⁾として提案されたのが最初のようですが、人工知能（artificial intelligence, AI）における確率推論アルゴリズムや統計物理分野のペーテ近似など、さまざまな分野で同等の手法が独立に開発されていました。確率伝播法は非常に強力で汎用性をもった手法で、現在でもさまざまな問題に適用されています。興味のある方は、スペースベクトル再構成^{109), 110)} や離散値ベクトル再構成⁵²⁾、平均合意¹¹¹⁾（average consensus）などへの適用例を参照してください。

A

付録：よく使う行列に関する 命題と性質

Next SIP

ここでは、通信のための信号処理で頻繁に利用される行列に関する補題や定理、性質について説明します。

A.1 逆行列補題

逆行列補題 (matrix inversion lemma) の式は一見複雑であまり使い道がないよう思えますが、実際には逆行列計算の演算量を削減する目的などでよく利用されています。

定理 A.1 (逆行列補題)¹²⁾ 行列 $\mathbf{A} \in \mathbb{C}^{N \times N}$ と $\mathbf{D} \in \mathbb{C}^{M \times M}$ は正則な正方形行列とする。一方、行列 $\mathbf{B} \in \mathbb{C}^{N \times M}$ と $\mathbf{C} \in \mathbb{C}^{M \times N}$ は必ずしも正方ではない行列とする。このとき、つぎの等式が成立する。

$$(\mathbf{A} - \mathbf{B}\mathbf{D}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}^{-1} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \quad (\text{A.1})$$

逆行列補題は、式 (A.1) の右辺の左あるいは右から $\mathbf{A} - \mathbf{B}\mathbf{D}\mathbf{C}$ を乗算するとそれが単位行列になることを示すことで証明できますが、以下のブロック行列を用いた証明法を用いると逆行列補題を自ら簡単に導出できるのでぜひ覚えておいてください。

【証明】 行列 $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ で構成される $(N + M) \times (N + M)$ のブロック行列 \mathbf{V} をつぎで定義します。

$$\mathbf{V} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \quad (\text{A.2})$$

ブロック行列 \mathbf{V} の逆行列はこれと同一の構造をもったブロック行列として書くことができます。すなわち

$$\mathbf{V}^{-1} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Z} & \mathbf{W} \end{bmatrix} \quad (\text{A.3})$$

とすると、 $\mathbf{X} \in \mathbb{C}^{N \times N}$, $\mathbf{W} \in \mathbb{C}^{M \times M}$, $\mathbf{Y} \in \mathbb{C}^{N \times M}$, $\mathbf{Z} \in \mathbb{C}^{M \times N}$ です。 $\mathbf{V}\mathbf{V}^{-1} = \mathbf{I}_{N+M}$ と $\mathbf{V}^{-1}\mathbf{V} = \mathbf{I}_{N+M}$ が成立するので

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Z} & \mathbf{W} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M \end{bmatrix} \quad (\text{A.4})$$

と

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Z} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M \end{bmatrix} \quad (\text{A.5})$$

が成立します。これらを展開して書くと

$$\mathbf{AX} + \mathbf{BZ} = \mathbf{I}_N \quad (\text{A.6})$$

$$\mathbf{AY} + \mathbf{BW} = \mathbf{0} \quad (\text{A.7})$$

$$\mathbf{CX} + \mathbf{DZ} = \mathbf{0} \quad (\text{A.8})$$

$$\mathbf{CY} + \mathbf{DW} = \mathbf{I}_M \quad (\text{A.9})$$

$$\mathbf{XA} + \mathbf{YC} = \mathbf{I}_N \quad (\text{A.10})$$

$$\mathbf{XB} + \mathbf{YD} = \mathbf{0} \quad (\text{A.11})$$

$$\mathbf{ZA} + \mathbf{WC} = \mathbf{0} \quad (\text{A.12})$$

$$\mathbf{ZB} + \mathbf{WD} = \mathbf{I}_M \quad (\text{A.13})$$

となります。式 (A.8) より

$$\mathbf{Z} = -\mathbf{D}^{-1}\mathbf{CX} \quad (\text{A.14})$$

となり、これを式 (A.6) に代入して

$$\mathbf{X} = (\mathbf{A} - \mathbf{BD}^{-1}\mathbf{C})^{-1} \quad (\text{A.15})$$

を得ます。一方、式 (A.12) より

$$\mathbf{Z} = -\mathbf{WCA}^{-1} \quad (\text{A.16})$$

が成り立つので、これを式 (A.13) に代入して

$$\mathbf{W} = (\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B})^{-1} \quad (\text{A.17})$$

となり、さらにこれを式 (A.12) に戻すと

$$\mathbf{Z} = -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \quad (\text{A.18})$$

となります。最後にこれを式 (A.6) に代入すると

$$\mathbf{X} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \quad (\text{A.19})$$

となり、式 (A.15) と式 (A.19) の右辺を等号で結ぶことで

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}$$

を得ます。この式で行列 \mathbf{D}^{-1} を改めて \mathbf{D} と置き直すと逆行列補題の式 (A.1) が得られることがわかります。



上では逆行列補題のみについて示しましたが、ここで用いたブロック行列について考える手法を用いると、行列に関するさまざまな有益な等式を得ることができます。例えば、式 (A.1) では \mathbf{D} が逆行列をもつことを前提にしていましたが、 \mathbf{D} が正則でない場合には

$$(\mathbf{A} - \mathbf{B}\mathbf{D}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{I}_M - \mathbf{D}\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{C}\mathbf{A}^{-1}$$

と書くことができます。また、シュアーブル行列 (Schur complement) を用いたブロック行列の分解なども可能です。興味のある方は、文献22) の付録 A や文献112) などを参照してください。

A.2 クロネッカーアイデア

本書では KR 積拡張アレー処理にクロネッカーアイデアが登場しましたが、それ以外にも画像を扱う際などさまざまな信号処理の場面に現れます。信号処理に役立つクロネッカーアイデアのいくつかの性質を以下に挙げます⁵⁾。

定理 A.2 (クロネッカーアイデアの性質)

- (i) 行列 $\mathbf{A} \in \mathbb{C}^{M \times N}$, $\mathbf{B} \in \mathbb{C}^{P \times Q}$, $\mathbf{C} \in \mathbb{C}^{N \times S}$, $\mathbf{D} \in \mathbb{C}^{Q \times T}$ に対して,
 $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{A}\mathbf{C} \otimes \mathbf{B}\mathbf{D}$ となる。
- (ii) $\mathbf{A} \in \mathbb{C}^{M \times M}$ と $\mathbf{B} \in \mathbb{C}^{N \times N}$ が正則ならば, $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ となる。
- (iii) $\mathbf{A} \in \mathbb{C}^{M \times M}$ の固有値を $\alpha_1, \alpha_2, \dots, \alpha_M$, 対応する固有ベクトルを $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ とし, $\mathbf{B} \in \mathbb{C}^{N \times N}$ の固有値を $\beta_1, \beta_2, \dots, \beta_N$, 対応する固有ベクトルを $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$ とする。このとき, $\mathbf{A} \otimes \mathbf{B}$ の MN 個の固有値は

$\alpha_i \beta_j, i = 1, 2, \dots, M, j = 1, 2, \dots, N$ となり、対応する固有ベクトルは $\mathbf{x}_i \otimes \mathbf{y}_j, i = 1, 2, \dots, M, j = 1, 2, \dots, N$ となる。

- (iv) $\mathbf{A} \in \mathbb{C}^{M \times M}, \mathbf{B} \in \mathbb{C}^{N \times N}$ に対し、 $\text{tr}(\mathbf{A} \otimes \mathbf{B}) = (\text{tr} \mathbf{A})(\text{tr} \mathbf{B})$ となる。
- (v) $\mathbf{A} \in \mathbb{C}^{M \times M}, \mathbf{B} \in \mathbb{C}^{N \times N}$ に対し、 $\det(\mathbf{A} \otimes \mathbf{B}) = (\det \mathbf{A})^N (\det \mathbf{A})^M$ となる。ただし、 $\det(\cdot)$ は行列式を表す。
- (vi) $\mathbf{A} \in \mathbb{C}^{M \times N}, \mathbf{B} \in \mathbb{C}^{N \times P}, \mathbf{C} \in \mathbb{C}^{P \times Q}$ に対し、 $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A}) \text{vec}(\mathbf{B})$ となる。

【証明】

(i) はクロネッカー積の定義どおり計算することで容易に確認できます。

(ii) は(i)の特別な場合として、 $\mathbf{C} = \mathbf{A}^{-1}, \mathbf{D} = \mathbf{B}^{-1}$ とすると

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{A}^{-1} \otimes \mathbf{B}^{-1}) = \mathbf{A}\mathbf{A}^{-1} \otimes \mathbf{B}\mathbf{B}^{-1} = \mathbf{I}_M \otimes \mathbf{I}_N = \mathbf{I}_{MN}$$

となることから確認できます。

(iii) も(i)の特別な場合として、 $\mathbf{C} = \mathbf{x}_i, \mathbf{D} = \mathbf{y}_j$ とすると

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{x}_i \otimes \mathbf{y}_j) = \mathbf{A}\mathbf{x}_i \otimes \mathbf{B}\mathbf{y}_j = \alpha_i \mathbf{x}_i \otimes \beta_j \mathbf{y}_j = \alpha_i \beta_j (\mathbf{x}_i \otimes \mathbf{y}_j)$$

となることから確認できます。

(iv) は、トレースが固有値の和で与えられることと(iii)の結果を用いることで

$$\text{tr}(\mathbf{A} \otimes \mathbf{B}) = \sum_i \sum_j \alpha_i \beta_j = \left(\sum_i \alpha_i \right) \left(\sum_j \beta_j \right) = (\text{tr} \mathbf{A})(\text{tr} \mathbf{B})$$

により得られます。

(v) は、行列式が固有値の積で与えられることと(iii)の結果を用いることで

$$\begin{aligned} \det(\mathbf{A} \otimes \mathbf{B}) &= \prod_{i=1}^M \prod_{j=1}^N \alpha_i \beta_j = \left(\prod_{i=1}^M \alpha_i^N \right) \left(\prod_{j=1}^N \beta_j^M \right) \\ &= (\det \mathbf{A})^N (\det \mathbf{A})^M \end{aligned}$$

が得られます。

(vi) は、 $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \cdots \ \mathbf{b}_P], \mathbf{C} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \cdots \ \mathbf{c}_Q], \mathbf{c}_i = [c_{1,i} \ c_{2,i} \ \cdots \ c_{P,i}]^\top, i = 1, 2, \dots, Q$ とすると、行列 \mathbf{ABC} の j 番目の列ベクトルが \mathbf{ABC}_j で与えられることを利用します。これは

$$\begin{aligned}
 \underbrace{\mathbf{ABC}_j}_{\text{カッコ入れ?}} &= \mathbf{A} \sum_{i=1}^P \mathbf{b}_i c_{ij} = [\mathbf{c}_{1,j} \mathbf{A} \ \mathbf{c}_{2,j} \mathbf{A} \ \cdots \ \mathbf{c}_{P,j} \mathbf{A}] \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_P \end{bmatrix} \\
 &= (\mathbf{c}_j^\top \otimes \mathbf{A}) \operatorname{vec}(\mathbf{B})
 \end{aligned}$$

と変形できるので、これをスタックすることで

$$\operatorname{vec}(\mathbf{ABC}) = \begin{bmatrix} \mathbf{ABC}_1 \\ \mathbf{ABC}_2 \\ \vdots \\ \mathbf{ABC}_Q \end{bmatrix} = \begin{bmatrix} \mathbf{c}_1^\top \otimes \mathbf{A} \\ \mathbf{c}_2^\top \otimes \mathbf{A} \\ \vdots \\ \mathbf{c}_Q^\top \otimes \mathbf{A} \end{bmatrix} \operatorname{vec}(\mathbf{B}) = (\mathbf{C}^\top \otimes \mathbf{A}) \operatorname{vec}(\mathbf{B})$$

が得られます。 ◇

A.3 ゲルシュゴリンの定理

信号処理システムのさまざまな振舞いは、多くの場合そこに登場する行列の固有値によって決定づけられます。固有値分解の計算量は一般に大きいため、元の行列の成分を用いて簡単に固有値の大きさを見積ももることができると大変便利です。ゲルシュゴリンの定理 (Gershgorin circle theorem) はそのような要求に応えてくれます。

定理 A.3 (ゲルシュゴリンの定理)⁵⁾ $\mathbf{A} \in \mathbb{C}^{N \times N}$ の (i, j) 成分を $a_{i,j}$ とする。このとき \mathbf{A} の固有値は、 $a_{i,i}$, $i = 1, 2, \dots, N$ を中心、 $\sum_{k \neq i} |a_{i,k}|$ を半径とする複素平面上の N 個の円内もしくは円上に存在する。

【証明】 行列 \mathbf{A} のある固有値を $\lambda \in \mathbb{C}$ とし、 λ に対応する固有ベクトルを $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^\top \in \mathbb{C}^N$ とすると

$$\mathbf{Ax} = \lambda \mathbf{x} \quad (\text{A.20})$$

が成り立ちます。ここで、 \mathbf{x} は i 番目の成分の絶対値 $|x_i|$ が最大であるとして、式 (A.20) から i 番目の行の等式を抜き出して書くと

$$\sum_{k=1}^N a_{i,k} x_k = \lambda x_i \quad (\text{A.21})$$

となり、左辺の和から $k = i$ の成分だけを抜き出して移項すると

$$(\lambda - a_{i,i})x_i = \sum_{k \neq i} a_{i,k}x_k \quad (\text{A.22})$$

と書けます。つぎに両辺の絶対値をとり、右辺の和の絶対値を、各項の絶対値の和で上から押さえると

$$|\lambda - a_{i,i}| |x_i| = \left| \sum_{k \neq i} a_{i,k}x_k \right| \leq \sum_{k \neq i} |a_{i,k}| |x_k| \quad (\text{A.23})$$

が得られます。さらに両辺を $|x_i| (> 0)$ で割ると

$$|\lambda - a_{i,i}| \leq \sum_{k \neq i} |a_{i,k}| \frac{|x_k|}{|x_i|} \leq \sum_{k \neq i} |a_{i,k}| \quad (\text{A.24})$$

となります（ここで、 $|x_k| \geq |x_i|$ を用いました）。式 (A.24) は円を表す不等式であり、固有値 λ と $a_{i,i}$ の複素平面上のユークリッド距離が $\sum_{k \neq i} |a_{i,k}|$ 以下であること、

言い換えると、固有値 λ が $a_{i,i}$ を中心とする半径 $\sum_{k \neq i} |a_{i,k}|$ の円内もしくは円上に存在することを意味します。

ここまででは、固有ベクトル \mathbf{x} の i 番目の成分の絶対値が最大であることを前提に議論を進めましたが、固有値分解の計算を実際に行うことなく固有値の大きさを見積もりたいという場面では、当然、固有ベクトルのどの成分の絶対値が最大であるかは不明です。ただし、固有ベクトルのいずれかの成分の絶対値が最大になることはわかっているので、 $i = 1, 2, \dots, N$ について $a_{i,i}$ を中心とする半径 $\sum_{k \neq i} |a_{i,k}|$ の円を考えると、この N 個の円の少なくとも一つの円内もしくは円上に固有値が存在します。さらに、この議論は行列 \mathbf{A} の任意の固有値に対応する固有ベクトルについて成立します。以上より、ゲルシュゴリンの定理が成立することが示されました。

行列 \mathbf{A} の固有値とその転置行列 \mathbf{A}^T の固有値は一致するので、 \mathbf{A}^T に対しても同様に固有値が存在し得る N 個の円盤（disk）の和集合の領域を計算し、これと \mathbf{A} の N 個の円盤の和集合との共通部分を考えることで、よりタイトな固有値の存在領域を求めるることができます。



引用・参考文献

- 1) 山本 裕：システムと制御の数学（システム制御情報ライブラリー 16），朝倉書店（1998）
- 2) 佐武一郎：線型代数学，裳華房（1974）
- 3) 田中聰久：信号・データ処理のための行列とベクトル—複素数，線形代数，統計学の基礎—（次世代信号情報処理シリーズ 1），コロナ社（2019）
- 4) 山本直樹：複素関数論の基礎，裳華房（2015）
- 5) G. H. Golub and C. F. Van Loan : Matrix Computations, 4th Edition, The Johns Hopkins University Press (2013)
- 6) 藤澤洋徳：確率と統計（現代基礎数学 13），朝倉書店（2006）
- 7) W. Feller : An Introduction to Probability Theory and Its Applications Vol. 1, 3rd Edition, John Wiley & Sons (1968)
- 8) 森口繁一：確率表現関数，東京大学出版会（1995）
- 9) C. M. Bishop : Pattern Recognition and Machine Learning, Springer (2006)
- 10) W. A. Gardner : Cyclostationarity in Communications and Signal Processing, IEEE Press (1994)
- 11) 町田東一，駒崎友和，松浦武信：マトリクスの固有値と対角化，東海大学出版会（1990）
- 12) S. Haykin : Adaptive Filter Theory, 3rd Edition, Prentice Hall (1996)
- 13) B. F.-Boroujeny : Adaptive Filters, Theory and Applications, John Wiley & Sons (1998)
- 14) 渡辺澄夫，村田 昇：確率と統計—情報学への架橋—，コロナ社（2005）
- 15) 高橋幸雄：確率論（基礎数理講座 2），朝倉書店（2008）
- 16) 小林昭七：微分積分読本 1 変数，裳華房（2000）
- 17) 小林昭七：統 微分積分読本 多変数，裳華房（2001）
- 18) A. Hjørungnes : Complex-Valued Matrix Derivatives, Cambridge University Press (2011)
- 19) P. J. Schreier and L. L. Scharf : Statistical Signal Processing of Complex-Valued Data, Cambridge University Press (2010)

- 20) 小島紀男, 矢沢志雄作, 本間光一:マトリクスとシステム, 東海大学出版会(1990)
- 21) L. Schwartz:複素関数(シュヴァルツ解析学6), 東京図書(1971)
- 22) T. Kailath, A. Sayed, and B. Hassibi: Linear Estimation, Prentice Hall (2000)
- 23) 黒田成俊:微分積分(共立講座21世紀の数学1), 共立出版(2002)
- 24) 片山 徹:新版応用カルマンフィルタ, 朝倉書店(2000)
- 25) T. M. Cover and J. A. Thomas: Elements of Information Theory, 2nd Edition, Wiley-Interscience (2006)
- 26) D. Tse and P. Viswanath: Fundamentals of Wireless Communication, Cambridge University Press (2005)
- 27) S. Moshan: Multi-user Detection for DS-CDMA Communications, IEEE Commun. Mag., **34**, 10, pp.124–136 (1996)
- 28) A. Chockalingam and B. S. Rajan: Large MIMO Systems, Cambridge University Press (2014)
- 29) D. P. Bertsekas and J. N. Tsitsiklis: Introduction to Probability, Athena Scientific (2008)
- 30) A. J. Viterbi: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Trans. Inf. Theory, **IT-13**, pp.260–269 (1967)
- 31) G. D. Forney: The Viterbi Algorithm, Proc. IEEE, **61**, 3, pp.268–278 (1973)
- 32) B. Hassibi and H. Vikalo: On the Sphere-Decoding Algorithm I. Expected Complexity, IEEE Trans. Signal Process., **53**, 8, pp.2806–2818 (2005)
- 33) R. M. Gray: Toeplitz and Circulant Matrices: A review, Now Pub (2006)
- 34) P. J. Davis: Circulant Matrices, John Wiley & Sons (1979)
- 35) O. Edfors, M. Sandell, J. J. van de Beek, S. K. Wilson, and P. O. Borjesson: OFDM Channel Estimation by Singular Value Decomposition, IEEE Trans. Commun., **46**, 7, pp.931–939 (1998)
- 36) K. Hayashi, M. Nagahara, and T. Tanaka: A User's Guide to Compressed Sensing for Communications Systems, IEICE Trans. Commun., **E96-B**, 3, pp.685–712 (2013)
- 37) 大鐘武雄, 小川恭孝: わかりやすいMIMOシステム技術, オーム社(2009)
- 38) R. Hayakawa and K. Hayashi: Reconstruction of Complex Discrete-Valued Vector via Convex Optimization with Sparse Regularizers, IEEE Access, **6**, 1, pp.66499–66512 (2018)

- 39) D. L. Donoho : Compressed Sensing, *IEEE Trans. Inf. Theory*, **52**, 4, pp.1289–1306 (2006)
- 40) E. J. Candes and T. Tao : Decoding by Linear Programming, *IEEE Trans. Inf. Theory*, **51**, 12, pp.4203–4215 (2005)
- 41) E. J. Candes, J. Romberg, and T. Tao : Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information, *IEEE Trans. Inf. Theory*, **52**, 2, pp.489–509 (2006)
- 42) M. Sipser : *Introduction to the Theory of Computation*, 3rd Edition, Cengage Learning (2012)
- 43) R. Tibshirani : Regression Shrinkage and Selection via the Lasso, *J. R. Statist. Soc. B*, **58**, 1, pp.267–288 (1996)
- 44) 小野嶽佑:近接分離アルゴリズムとその応用, *Communications of the Operations Research Society of Japan*, **64**, 6, pp.316–325 (2019)
- 45) P. L. Combettes and J.-C. Pesquet : Proximal Splitting Methods in Signal Processing, in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer (2011)
- 46) P. L. Combettes and V. Wajs : Signal recovery by proximal forward-backward splitting, *Multiscale Modeling and Simulation*, **4**, pp.1168–1200 (2005)
- 47) A. Beck and M. Teboulle : A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, *SIAM J. Image Science*, **2**, 1, pp.183–202 (2009)
- 48) J. Eckstein and D. P. Bertsekas : On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, **55**, pp.293–318 (1992)
- 49) M. Nagahara : Discrete Signal Reconstruction by Sum of Absolute Values, *IEEE Signal Process. Lett.*, **22**, 10, pp.1575–1579 (2015)
- 50) H. Sasahara, K. Hayashi, and M. Nagahara : Multiuser Detection based on MAP Estimation with Sum-of-Absolute-Values Relaxation, *IEEE Trans. Signal Process.*, **65**, 21, pp.5621–5634 (2017)
- 51) R. Hayakawa and K. Hayashi : Convex Optimization Based Signal Detection for Massive Overloaded MIMO Systems, *IEEE Trans. Wireless Commun.*, **16**, 11, pp.7080–7091 (2017)
- 52) R. Hayakawa and K. Hayashi : Discreteness-Aware Approximate Message

- Passing for Discrete-Valued Vector Reconstruction, IEEE Trans. Signal Process., **66**, 24, pp.6443–6457 (2018)
- 53) M. Elad : Sparse and Redundant Representations, Springer (2010)
- 54) Y. C. Eldar and G. Kutyniok : Compressed Sensing, Cambridge University Press (2012)
- 55) N. Sidiropoulos and R. Bro : PARAFAC Techniques for Signal Separation, in Signal Processing Advances in Wireless and Mobile Communications, **2**, pp. 131–179, Prentice Hall (2001)
- 56) E. J. Candes : The Restricted Isometry Property and its Implications for Compressed Sensing, Comptes Rendus de l'Academie des Sciences, Paris, Ser. I, **346**, 9-10, pp.589–592 (2008)
- 57) A. Cohen, W. Dahmen, and R. DeVore : Compressed sensing and best k -term approximation, J. Amer. Math. Soc., **22**, 1, pp.211–231 (2009)
- 58) 永原正章：スパースモデリング—基礎から動的システムへの応用—, コロナ社 (2017)
- 59) A. V. Oppenheim and R. W. Schafer : Discrete-Time Signal Processing, 3rd Edition, Prentice Hall (2010)
- 60) 酒井英昭：主成分分析と独立成分分析, システム／制御／情報, **43**, 4, pp.188–195 (1999)
- 61) H. Hotelling : Analysis of a Complex of Statistical Variables into Principal Components, Journal of Educational Psychology, **24**, pp. 417-441 (1933)
- 62) I. T. Jolliffe : Principal Component Analysis, Springer (2002)
- 63) R. O. Schmidt : Multiple Emitter Location and Signal Parameter Estimation, IEEE Trans. Antennas and Propag., **AP-34**, 3, pp.276–280 (1986)
- 64) B. D. Rao and K. V. S. Hari : Performance Analysis of Root-Music, IEEE Trans. Acoust., Speech, Signal Process., **37**, 12, pp.1939–1949 (1989)
- 65) T-J. Shan, M. Wax, and T. Kailath : On Spatial Smoothing for Direction-of-Arrival Estimation of Coherent Signals, IEEE Trans. Acoust., Speech, Signal Process., **33**, 4, pp.806–811 (1985)
- 66) R. T. Williams, S. Prasad, A. K. Mahalanabis, and L. H. Sibul : An Improved Spatial Smoothing Technique for Bearing Estimation in a Multipath Environment, IEEE Trans. Acoust., Speech, Signal Process., **36**, 4, pp.425–432 (1988)
- 67) S. U. Pillai and B. H. Kwon : Forward/Backward Spatial Smoothing Tech-

- niques for Coherent Signal Identification, IEEE Trans. Acoust., Speech, Signal Process., **37**, 1, pp.8–15 (1989)
- 68) R. Roy and T. Kailath : Estimation of Signal Parameters Via Rotational Invariance Techniques, IEEE Trans. Acoust., Speech, Signal Process., **37**, 7, pp.984–995 (1989)
- 69) 菊間信良 : アダプティブアンテナ技術, オーム社 (2003)
- 70) W.-K. Ma, T.-H. Hsieh, and C.-Y. Chi : DOA Estimation of Quasi-Stationary Signals via Khatri-Rao Subspace, in Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP), pp.2165–2168 (2009)
- 71) W.-K. Ma, T.-H. Hsieh, and C.-Y. Chi : DOA Estimation of Quasi-Stationary Signals with Less Sensors Than Sources and Unknown Spatial Noise Covariance: A Khatri-Rao Subspace Approach, IEEE Trans. Signal Process., **58**, 4, pp.2168–218 (2010)
- 72) P. Pal and P. P. Vaidyanathan : Nested Arrays: A Novel Approach to Array Processing with Enhanced Degrees of Freedom, IEEE Trans. Signal Process., **58**, 8, pp.4167–4181 (2010)
- 73) W.-T. Zhang and S.-T. Lou : Search Free Algorithms for DOA Estimation of Quasi-Stationary Signals, in Proc. IEEE Int. Workshop on Machine Learning for Signal Process. (MLSP), pp.1–5 (2011)
- 74) M. Inoue, K. Hayashi, H. Mori, and T. Nabetani : A DOA Estimation Method with Kronecker Subspace for Coherent Signals, IEEE Communications Letters, **22**, 11, pp.2306–2309 (2018)
- 75) H. L. Van Trees : Optimum Array Processing, Wiley-Interscience (2002)
- 76) P. Stoica and A. Nehorai : MUSIC, Maximum Likelihood, and Cramer-Rao Bound, IEEE Trans. Acoust., Speech, Signal Process., **37**, 5, pp.720–741 (1989)
- 77) 唐沢好男 : 高分解能到来方向推定の仕組み Part I : これでわかる MUSIC 法と ESPRIT 法, TR-YK-049 (2020)
- 78) 唐沢好男 : 高分解能到来方向推定の仕組み Part II : コヒーレント波が入射する場合の F/B 型空間平均法, TR-YK-050 (2020)
- 79) J. M. Kim, O. K. Lee, and J. C. Ye : Compressive MUSIC: Revisiting the Link Between Compressive Sensing and Array Signal Processing, IEEE Trans. Inf. Theory, **58**, 1, pp.278–301 (2012)
- 80) H. Mukumoto, K. Hayashi, and M. Kaneko : Direction-Of-Arrival Estima-

- tion via Khatri-Rao Subspace Using Compressed Sensing, in Proc. Asia-Pacific Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), Jeju, Korea (2016)
- 81) E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue : Subspace Methods for the Blind Identification of Multichannel FIR Filters, IEEE Trans. Signal Process., **43**, 2, pp.516–525 (1995)
- 82) K. Hayashi, H. Matsushima, H. Sakai, E. Carvalho, and P. Popovski : Subspace Based Blind Sparse Channel Estimation, in Proc. Asia-Pacific Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC) , pp.1–6 (2012)
- 83) G. Xu and T. Kailath : Direction-of-Arrival Estimation via Exploitation of Cyclostationarity - A Combination of Temporal and Spatial Processing, IEEE Trans. Signal Process., **40**, 7, pp.1775–1786 (1992)
- 84) L. Tong, G. Xu, and T. Kailath : Blind Identification and Equalization based on Second-Order Statistics: A Time Domain Approach, IEEE Trans. Inf. Theory, **40**, 2, pp.340–349 (1994)
- 85) L. Tong, G. Xu, B. Hassibi, and T. Kailath : Blind Channel Identification based on Second-Order Statistics : A Frequency Domain Approach, IEEE Trans. Inf. Theory, **40**, 1 pp.329–334 (1995)
- 86) 北川源四郎 : 時系列解析入門, 岩倉書店 (2005)
- 87) 樋口知之, 上野玄太, 中野慎也, 中村和幸, 吉田 亮 : データ同化入門—次世代のシミュレーション技術—(シリーズ予測と発見の科学 6), 朝倉書店 (2011)
- 88) 樋口知之 : 粒子フィルタ, 電子情報通信学会誌, **88**, 12, pp.989–994 (2005)
- 89) N. J. Gordon, D. J. Salmond, and A. F. M. Smith : Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, IEE Proceedings F, **140**, 2, pp.107–113 (1993)
- 90) G. Kitagawa : Monte Carlo Filtering and Smoothing for Nonlinear non-Gaussian State Space Models, Journal of Computational and Graphical Statistics, **5**, 1, pp.1–25 (1996)
- 91) 田中和之 : 結合ガウス・マルコフ確率場モデルに対するクラスター変分法による統計力学的反復計算アルゴリズム, 人工知能学会論文誌, **16**, 2, pp.259–267 (2001)
- 92) S. M. Aji and R. J. McEliece : The Generalized Distributive Law, IEEE Trans. Inf. Theory, **46**, 2, pp.325–343 (2000)

- 93) T. J. Richardson and R. L. Urbanke : The Capacity of Low-Density Parity Check Codes under Message-Passing Decoding, IEEE Trans. Inf. Theory, **47**, 2, pp.599–618 (2001)
- 94) S. Y. Chung, T. J. Richardson, and R. L. Urbanke: Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes using a Gaussian Approximation, IEEE Trans. Inf. Theory, **47**, 2, pp.657–686 (2001)
- 95) S. Brink : Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes, IEEE Trans. Commun., **49**, 10, pp.1727–1737 (2001)
- 96) 池田思朗, 田中利幸, 甘利俊一: 確率伝搬法の情報幾何—符号理論, 統計物理, 人工知能の接点—, 応用数理, **14**, 3, pp.24–35 (2004)
- 97) 池田思朗, 田中利幸, 岡田真人: 「確率的情報処理としての移動体通信技術」講義ノート, 若手研究者・学生向けに最新技術をわかりやすく紹介する講演会 (2002)
- 98) J. S. Yedidia, W. T. Freeman, and Y. Weiss : Bethe Free Energy, Kikuchi Approximations, and Belief Propagation Algorithms, (<http://www.merl.com/papers/TR2001-16>)
- 99) J. S. Yedidia, W. T. Freeman, and Y. Weiss : Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms, IEEE Trans. Inf. Theory, **51**, 7, pp.2282–2312 (2005)
- 100) R. G. Gallager : Low-Density Parity-Check Codes, MIT Press (1963)
- 101) J. Pearl : Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann Publishers Inc. (1988)
- 102) D. J. C. MacKay : Good Error-Correcting Codes based on Very Sparse Matrices, IEEE Trans. Inf. Theory, **45**, 2, pp.399–431 (1999)
- 103) 和田山 正: 誤り訂正技術の基礎, 森北出版 (2010)
- 104) C. Berrou, A. Glavieux, and P. Thitimajshima : Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1, Proc. of IEEE International Conference on Communications, **2**, pp.1064–1070 (1993)
- 105) R. J. McEliece, D. J. C. MacKay, and J. Cheng : Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm, IEEE J. Sel. Areas Commun., **16**, 2, pp.140–152 (1998)
- 106) J. W. Cooley and J. W. Tukey : An Algorithm for the Machine Calculation of Complex Fourier Series, Mathematics of Computation, **19**, 90, pp.297–301 (1965)
- 107) M. Heideman, D. Johnson and C. Burrus : Gauss and the history of the fast

- Fourier transform, IEEE ASSP Mag., **1**, 4, pp.14-21 (1984)
- 108) F. R. Kschischang, B. J. Frey, and H. A. Loeliger : Factor Graphs and the Sum-Product Algorithm, IEEE Trans. Inf. Theory, **47**, 2, pp.498-519 (2001)
- 109) D. L. Donoho, A. Maleki, and A. Montanari : Message-Passing Algorithms for Compressed Sensing, Proc. of the National Academy of Sciences (PNAS), **106**, 45, pp.18914-18919 (2009)
- 110) D. L. Donoho, A. Maleki, and A. Montanari : Message Passing Algorithms for Compressed Sensing: I. Motivation and Construction, in Proc. IEEE Inf. Theory Workshop, pp.1-5 (2010)
- 111) C. C. Moallemi and B. Van Roy : Consensus Propagation, IEEE Trans. Inf. Theory, **52**, 11, pp.4753-4766 (2006)
- 112) H. V. Henderson and S. R. Searle : On Deriving the Inverse of a Sum of Matrices, SIAM Review, **23**, pp.53-60 (1981)

索引

【あ】		【お】		【き】	
悪条件	112	オラクル推定量	112	機械学習	80
圧縮可能信号	98			木構造	173
圧縮センシング	98			期待値	13
誤り確率	81	回転不变性	141	——をとる操作	13
アルファベット	80	外部値	189	擬定常	148
アレー信号処理	119	ガウス近似法	180	擬定常性	143
		ガウス分布	67	希望信号	62
		ガウス放送通信路	74	逆行列	61
		角周波数	119	逆行列補題	67, 199
位相差	121	確率過程	19	逆問題	56
一様分布	81	確率質量関数	11	逆離散フーリエ変換	90
一般化確率伝播法	180	確率推論	11, 103	球面復号化	82
一般(化)状態空間モデル		確率的勾配降下法	55	狭義定常過程	20
	153	確率伝播法	170	狭帯域信号	119
入れ子型アレー	147	確率分布	8	強定常過程	20
インコヒーレント	122	確率変数	1	共分散	15
因子分解	31	確率密度関数	11	共分散行列	18
因数分解	31	隠れマルコフモデル	153	行ベクトル	41
インパルス応答	83	ガード区間	87	共役転置	5
		カーネル空間	126	行列式	1
		加法性白色ガウス雑音	82	極小値	41
ヴァンデルモンド行列	131	加法性白色雑音ベクトル	56	虚数単位	2
ウィルティンガー微分	39	加法定理	9	虚部	2
		カルマンフィルタ	164	近似的にスパースな信号	98
		ガロア体	184	近接勾配法	105
エピグラフ	103	完全再構成	110	近接写像	103
エルミート行列	5	観測行列	56	近接分離	103
エルミート形式	23	観測雑音	154		
エルミート転置	5	観測雑音ベクトル	56		
エルミート平方根行列	137	観測モデル	152	空	103
円盤	204	緩和	101	鎖状グラフィカルモデル 153	→ A
				クラスタリング	198

クラス分類	80	最尤系列推定	92	受信アンテナ	82		
グラフィカルモデル	31	最尤推定	81	主成分分析	128		
グラム行列	80	雑音強調	63	巡回行列	89		
クロネッカー積	144	雑音除去	127	準ノルム	7		
【け】							
経路	82	雑音部分空間	123	条件数	112		
結合確率	9	差分アレー	147	条件付き確率	9		
結合確率分布	9	三角不等式	7	条件付き確率分布	9		
ゲルシュゴリンの定理		サンプリング定理	122	条件付き期待値	64		
	117, 203	サンプル相関行列	127	条件付き共分散行列	165		
減算型干渉除去	72	【し】			条件付き独立性	30	
【こ】						状態	151
広義定常過程	20	時系列	19	状態推定	151		
交互方向乗数法	108	指向性	131	情報幾何	180		
高速フーリエ変換	86, 196	事後確率	80	乗法定理	9		
勾配ベクトル	40	自己共分散関数	20	シングルキャリヤブロック			
勾配法	41	自己共分散行列	22	伝送方式	91		
硬判定値	74	自己相關関数	19	信号検出	80		
行路差	120	自己相関行列	21	信号対干渉プラス雑音			
コーチー・シュワルツの		指示関数	104	電力比	68		
不等式	99	システム雑音	154	信号対雑音電力比	59		
コーチー・リーマンの		システムモデル	152	人工知能	198		
方程式	46	事前確率	81	信号部分空間	123		
コスト関数	41	実現値	1	信号プラス雑音部分空間	127		
コヒーレント	122	実効定義域	103	シンボル	80		
固有値	24	実数	1	シンボル間干渉	83		
固有ベクトル	24	実微分可能	46	シンボル周期	83		
固有方程式	78	実部	2	【す】			
【さ】						弱定常過程	20
サイクリックプレフィックス		周期定常過程	21	推定誤差	64		
クス	89	集合	1	隨伴行列	5		
サイクル	173	重点サンプリング	161	ステアリングベクトル	131		
最小ノルム解	96	周波数応答	83	スナップショット	121		
最小分散推定値	66	周波数推定問題	120	スパーク	110		
最小平均2乗誤差推定	63	周波数選択性通信路	83	スペース	98		
最小2乗推定	61	周波数非選択性通信路	83	スペースモデリング	118		
最大事後確率推定	81	周波数領域等化	86	スペニング木	197		
最大周辺事後確率推定	169	十分条件	111	【せ】			
最大比合成	77	周辺化	10	正規直交基底	28		
最適	81	周辺事後分布	30	制限等長性	111		
		縮退している	169	整合フィルタ推定	80		
			27	生成行列	185		
				正則化	96		

A

正則化項	97	多項式時間	101	等式制約	102
正則化最小2乗法	97	疊込み行列	82	到来方向	119
正則関数	45	タナーグラフ	186	等利得合成	77
正則行列	129	多変数関数	36	特異値	63
正定値	23	ターボ符号	187	特異値分解	62
成分ごと最大事後確率推定				独立	10
制約条件	59	【ち】		独立同分布	80
制約付き最小化問題	60	運延時間	122	凸関数	41
絶対値	3	運延広がり	83	凸緩和問題	101
零行列	1	逐次干渉除去	72	凸集合	41
零空間特性	114	チャネル硬化	80	トレース	1
線形・ガウス型状態空間		重複符号化	74	トレーニング信号	127
モデル	154	重複度	126	貪欲法	103
線形観測モデル	56	直交基底	27	【の】	
線形逆問題	57	直交射影	128	ノルム	6
線形計画問題	101	直交補空間	126		
線形推定	58	【つ】		【は】	
線形性	14	通信路応答	92	背理法	110
センサ	120	通信路推定	92	パイロット信号	92
センサアレー	120	通信路等化	84	波 源	120
センサ間隔	122	通信路容量	187	波長分散	83
選択合成	76	通信路容量領域	74	パラメータ	41
全微分	42	【て】		パリティ検査行列	185
【そ】		提案分布	161	半正定値	23
相 関	15	定義域	39	搬送波周波数	122
相関行列	17	低密度パリティ検査		【ひ】	
相互コヒーレンス	116		181, 184	非線形・非ガウス型状態	
送信アンテナ	82	停留点	41	空間モデル	153
送信等化	85	データ行列	148	比帶域	123
測度論	38	データ忠実	97	ビタビアルゴリズム	82
組織符号	187	テブリツツ行列	22	左特異ベクトル	149
ソフトしきい値作用素	105	テンソル分解	110	必要条件	110
【た】		転 置	4	非定常過程	21
台	7	電波干渉計	139	非凸最適化問題	101
帯域制限	122	【と】		非凸性	100
大域的な最小値	41	等位相面	120	非負定値	23
対角行列	1	等価低域系	121	微分可能	40, 106
大規模 MIMO	80	統計量	13	ビームフォーミング	131
対称行列	5	同時確率	9	【ふ】	
たがいに独立	10	同時確率分布	9	ファクターグラフ	36

フィルタ分布	155	平面波近似	120	有向非巡回グラフ	31	
複素共役	2	並列干渉除去	72	尤度関数	81	
複素振幅	119	並列連接組織符号	187	ユークリッドノルム	6	
複素数	1	ベクトル化	144	ユニタリ行列	27	
複素正弦波	119	ベクトル空間	6	ユニタリ相似変換	89	
複素微分可能な関数	45	ペーテ自由エネルギー	180	【よ】		
複素包絡線	123	ベルヌーイ分布	114	予測分布	155	
符号化率	185	変調信号	122	【ら】		
符号関数	99	偏波モード分散	83	【り】		
符号語	185	偏微分	40	ラグランジュ関数	60	
符号長	185	【ま】		ラグランジュ乗数	60	
符号分割多元接続	72	マイナー成分分析	129	ラグランジュ未定乗数法	60	
負定値	23	マルコフ性	152	【る】		
不等式制約	102	マルコフ連鎖	32	離散コサイン変換	114	
部分空間	27	マルチチャネル信号処理	150	離散性	100	
部分空間法	119	マルチバス通信路	82	離散フーリエ変換	89	
ブラインド同定法	150	マルチユーザ検出法	72	リサンプリング	162	
フレーム	148	【み】		リブシッツ連続	106	
プロック間干渉	87	密度発展法	180	粒子	159	
プロックごと最大事後確率		【む】		粒子フィルタ	158	
推定	169	無記憶	188	良条件	112	
プロバー	103	無指向性	120	【れ】		
分散	14	無相関	15	累積分布関数	12	
分散共分散行列	19	【め】		【れ】		
分配法則	170	モデル化誤差	98	レイリー限界	130	
【へ】		97	【も】		レイリー商	25
平滑化分布	155	メッセージ伝播	36	劣化型放送通信路	74	
平均合意	198	モデル化誤差	98	劣決定	96	
平均2乗誤差	64	【わ】		列フルランク	62	
閉形式	97	BP	170	列ベクトル	1, 41	
ペイジアンネットワーク	31	【ゆ】				
ペイズの定理	10	有界	102			
閉凸関数	103					
閉凸集合	103					

[A]

ADMM
AI
AWGN

[B]

BP
CDMA
CP

[C]

12
72
89



—を用いたシングル キャリヤブロック伝送 方式	91	【K】 Khatri-Rao 積 KR 積 k-ランク	143 143 110	PIC PMF prox 可能	72 11 106
【D】 DAG DCT DFT DOA	31 114 89 119	【L】 ℓ_0 再構成 Lasso LDPC	100 102 181, 184	RIP Rx. antenna SC SC-CP	111 82 76 91
【E】 EGC ESPRIT 法 EXIT チャート法	77 139 180	【M】 MAP 推定 MF 推定 ML 推定 MLSE	81 80 81 92	SGD SIC SINR SIR SNR	55 72 68 159 59
【F】 FDE FFT FISTA	86 86, 196 107	【N】 MMSE 推定 MPM 推定 MRC MUSIC スペクトル MUSIC 法	63 169 77 132 130	SOAV 最適化 SVD Tx. antenna	109 62 82
【G】 GF	184	【O】 NP 困難 NSP	101 114	ZF 推定	58
【H】 HMM	153	【P】 OFDM PCA PDF	91 128 11	【数字】 2 次過程 2 次形式 σ-集合体	20 23 38
【I】 IBI IDFT ISI ISTA i.i.d.	87 90 83 107 80	【ギリシャ文字】			

——監修者・著者略歴——

田中 聰久（たなか としひさ）
 1997年 東京工業大学工学部電気・電子工学科
 卒業
 2000年 東京工業大学大学院理工学研究科修士
 課程修了
 2002年 東京工業大学大学院理工学研究科博士
 後期課程修了、博士（工学）
 2002年 理化学研究所脳科学総合研究センター
 研究員
 2004年 東京農工大学講師
 2006年 東京農工大学助教授
 2007年 東京農工大学准教授
 2018年 東京農工大学教授
 現在に至る

林 和則（はやし かずのり）
 1997年 大阪大学工学部通信工学科卒業
 1999年 大阪大学大学院工学研究科博士前期課
 程修了（電子情報エネルギー工学専攻）
 2002年 大阪大学大学院工学研究科博士後期課
 程修了（電子情報エネルギー工学専攻）、
 博士（工学）
 2002年 京都大学助手
 2007年 京都大学助教
 2009年 京都大学准教授
 2017年 大阪市立大学教授
 2020年 京都大学教授
 現在に至る

通信の信号処理

—線形逆問題、圧縮センシング、確率推論、ウィルティンガー微分—

Signal Processing for Communications

—Linear Inverse Problems, Compressed Sensing, Probabilistic Inference,
and Wirtinger Derivative—

© Kazunori Hayashi 2023

2023年11月2日 初版第1刷発行

検印省略

監修者	田 中 聰 久
著者	林 和 則
発行者	株式会社 コロナ社
	代表者 牛来真也
印刷所	三美印刷株式会社
製本所	有限会社 愛千製本所

112-0011 東京都文京区千石4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-01406-8 C3355 Printed in Japan

(斎藤)



<出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、
 出版者著作権管理機構（電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。
 購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。
 落丁・乱丁はお取替えいたします。