令和 4 年度　卒業論文

機械学習を用いた XRISM 衛星搭載極低温検出器の異常検知アルゴリズムの開発

横浜国立大学　理工学部　数物・電子情報系学科
物理工学教育プログラム
中村研究室

学籍番号 1964064

柏﨑未有
kashiwazaki-miu-rp@ynu.jp

2023 年 2 月 14 日

概要

衛星実験では，打ち上げ前後から切れ目無く様々な動作確認を行なわなければならない。そのためには，いち早くトラブルを捕捉する異常検知システムが必要不可欠である。2016 年に打ち上げられた X 線天文衛星 ASTRO-H は、事故により打ち上げから 1 ヶ月で運用を停止した。故にその後継機である XRISM 衛星には、衛星異常検知システムの導入は、特に重要と認識されている。

現在 XRISM 衛星で運用予定の異常検知器 ATMOS は、問題点がある。これは、ATMOS は一次処理時系列テレメトリーデータの閾値による異常検知を行ので、高次処理を施したノイズスペクトルに含まれる異常や、時系列データのパターンに隠される異常検出が行えないことだ。そこで ATMOS を補完する XRISM 衛星の異常検知器、特に衛星搭載の極低温検出器 *Resolve* 装置に関する異常検知器を、開発する必要がある。従来の方法では検出できない異常を検出するにあたり、今回は機械学習的手法を用いることで、検出が可能であるかを検討した。よって本論文の目的は、XRISM 衛星搭載極低温検出器 *Resolve* に対する異常検知システムの開発にあたり、機械学習的手法の有効性を示すこととした。

Resolve 装置の異常検知アルゴリズムの開発にあたり、テレメトリデータに現れる二種の異常検知について扱った。一つ目が、検出器ノイズスペクトルに含まれる異常である。*Resolve* 装置の高エネルギー分解能を保つためには、異常なノイズスペクトルを示す時間帯を除去し、検出器の性能を安定化する必要がある。ゆえに異常なノイズスペクトルを検知するアルゴリズムの開発を行った。二つ目が、検出器の温度時系列データに含まれる異常である。*Resolve* 装置の X 線検出器は、通常 50 mK に制御されている。しかし *Resolve* 装置に 50 mK を保持する冷却系に対し何かしらの動作が行われる場合か、異常が発生した場合は 50 mK から温度が外れる。そこで検出器が温度変化した時刻を検知し、温度変化した原因が意図しない異常なものであれば、それを検知する必要がある。今回はこの課題に対し、50 mK ステージの温度が温度変化をした時刻の検知を行い、検知時刻後の温度変化パターンから異常であるかを判断するアルゴリズム開発を目標とした。ただ今回は、50 mK ステージの温度の温度変化時刻を検地するアルゴリズムの開発までが完了した。

本研究では SGDClassifier や LinearSVC などの教師あり学習法と、Kmeans などの教師なし学習方法を含む機械学習アルゴリズムを中心にアルゴリズムを構築した。加えて時系列解析には、自己回帰モデルを用いた統計的手法等も組み込むことで、より精度の高い異常検知アルゴリズムを検討した。また、これらの開発された異常検知アルゴリズムは、地上試験の段階で *Resolve* によって収集された実際のデータを使用してテストされた。よってこれらのアルゴリズムの結果は、実際の運用時にも同様な結果が出ることが期待できる。

アルゴリズムの開発の結果、検出器ノイズスペクトルに含まれる異常に対しては、高い精度での異常検知ができた。だが検出器の温度時系列データに含まれる異常に対しては、温度変化時刻の検知は達成したが、異常温度変化と正常温度変化の分類器の作成には至らなかった。ゆえに異常温度変化検知に関しては、まだ改善が期待される結果となった。だがアルゴリズムの開発の過程で、機械学習を利用したアルゴリズムの有用性が実証されたので、実際の運用に使用できるシステムに組み込むのが今後の課題である。

目次

図目次

表目次

第 1 章

研究の設計

Contents

1.1 背景

1.1.1 ASTRO-H 衛星の事故

ASTRO-H 衛星とは、NASA や ESA との国際協力で、2008 年に開始され、2016 年に打ち上げられた、JAXA 宇宙科学研究所の 6 番目の X 線天文衛星である。ブラックホール、超新星残骸、銀河団など、X 線やガンマ線で観測される高温、高エネルギーの天体の研究を通じて、宇宙の構造とその進化の解明を目的とした衛星であった。ASTRO-H 衛星には、四種類の新型観測装置が搭載され、ASTRO-H 衛星以前に開発された X 線天文衛星「すざく」に比べ、エネルギー分解能での 10 倍の分光観測を実現し、科学的成果を引き出すことが可能とされた。

しかし、ASTRO-H 衛星は打ち上げから約一か月後の 3 月 26 日に姿勢制御系の不具合が生じ、その後 4 月 28 日に復旧が断念され、計画は終了した。姿勢制御系の不具合が生じた日は、観測に向けた初期機能の確認を行っており、あるX線天体に向け、全観測機器の試験観測をしていた。以下では 3 月 26 日に生じた姿勢制御系の不具合の、発生メカニズムを説明する。発生メカニズムや当時の観測計画などを図示したものが、図（1.1)、図（1.2）である。ただし、以下の時刻は JST で示す。

1. 3 月 26 日の 3 時ごろに、活動銀河核の観測を行うための姿勢変更運用を行った。
2. 姿勢変更運用後、姿勢制御系の想定とは異なる動作により、実際に衛星が回転してないのにかかわらず、姿勢制御系は衛星が回転していると自己判断した。その結果、回転を止めようとする向きに、リアクションホイールを作動させた。これにより衛星に姿勢異常が発生した。
3. 加えて、姿勢制御系が実行する、磁気トルカによる角運動量のアンローディングが、姿勢異常のため正常に動作しなかった。そのため、リアクションホイールに角運動量が蓄積し続けた。
4. 姿勢制御系がこの状況を危険と判断し、衛星を安全な状態とするためセーフホールドに移行し、スラスタを噴射した。この際、姿勢制御系の不適切なスラスタ制御パラメータにより、想定と異なる指示をスラスタに与えた。その結果、スラスタは想定と異なる噴射を行い、衛星の回転が加速した。
5. 衛星の想定以上の回転運動により太陽電池パドルなどの、回転状態で発生する力に対し構造的に弱い部分が

破断し、分離した。

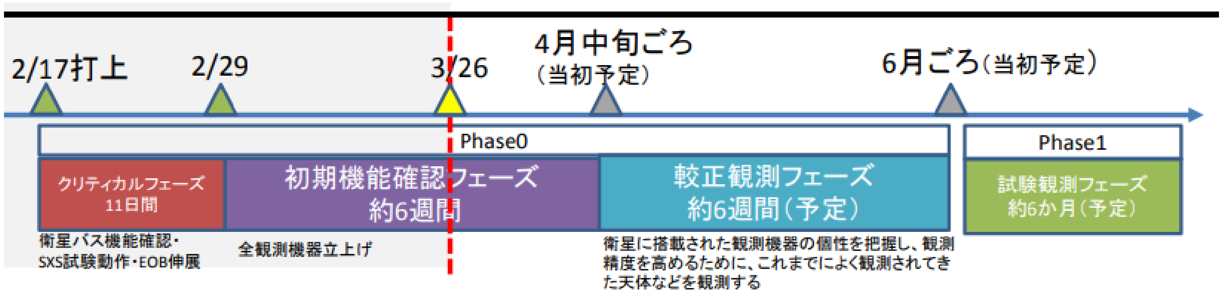

図 1.1. ASTRO-H 衛星の運用計画 (1)

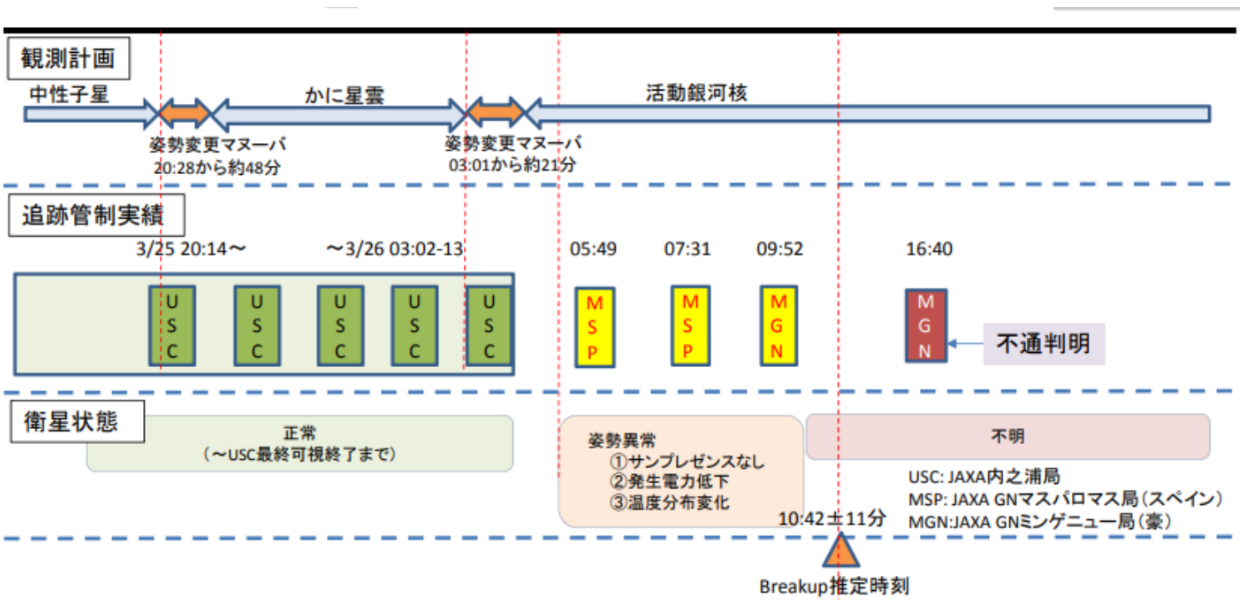

図 1.2. 異常発生時の状況 (1)

このような ASTRO-H 衛星の姿勢制御系の異常は、3 月 26 日の 3:00 ごろから始まり、同日 10:47 に太陽電池パドルなどが分離した。実際に衛星の状態を異常と判断したのは、10:00 ごろであった。3 月 26 日の、地上でのテレメトリデータの受信状況が図（1.3）である。図（1.3）をみると、テレメトリーデータから、衛星状態が異常であると検知したのは 10:00 頃であるが、姿勢制御系異常がすでに生じていた 6:00 の段階でテレメトリーデータは送られていた。この段階で衛星の状態が異常であることが検知され、異常に対応できる体制が組まれていたなら、衛星損失の回避をすることができた可能性がある。ASTRO-H 衛星の事故を踏まえ、今後の衛星では、同様の事故を防ぐため、衛星のテレメトリデータから、早期に衛星の異常を検知するシステムが必要と認識されることになった。

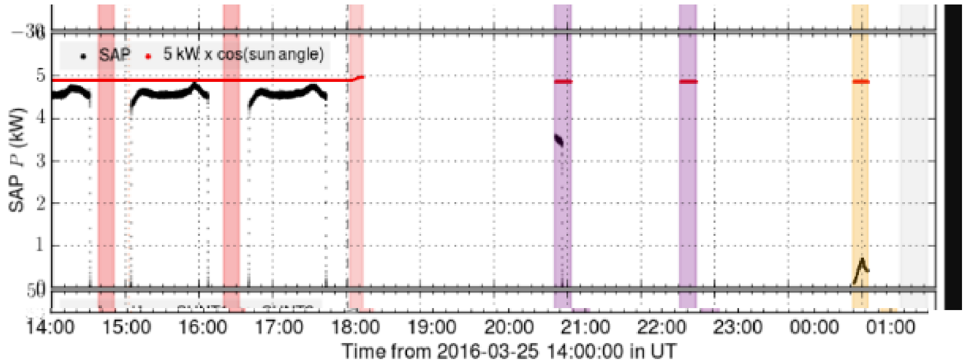

図 1.3. 異常発生時のテレメトリーデータ（時刻は UT）

1.1.2　衛星異常検知の試み

現在 XRISM 衛星には、ATMOS という衛星自動監視ソフトウエアが搭載されている。しかしこのシステムだけでは、XRISM 衛星の異常検知器として不十分である。ATMOS は、基本的にテレメトリの閾値判定のみに基づく異常検知を行う。そのため検出器ノイズなどの高次処理を施したデータの異常や、テレメトリの時系列パターンに潜む異常は見つけられないからだ。よって ATMOS を補完するような、異常検知器の開発が必要となる。

衛星異常検知器の開発に関しては、いくつかの先行研究がおこなわれている。(8),(10) 先行研究では機械学習や統計的手法を用いて、過去に打ち上げられた衛星の既知異常に対する、異常検知器開発が行われた。つまり実際に打ち上げられる衛星に対する、異常検知器ではない。ゆえに今回は、これらの先行研究で検討された異常検知アルゴリズムを踏まえ、実際に運用されている衛星、すなわち ASTRO-H 衛星の後継機である XRISM 衛星に対しての異常検知器の開発を行う。

1.2　目的

§1.1.1 で述べたように、ASTRO-H 衛星の事故を踏まえ、ASTRO-H 衛星の後継機である XRISM 衛星は、受信する衛星のテレメトリーデータから、衛星の異常を早期に検知する必要がある。そこで §1.1.2 で述べたように、過去の衛星の既知異常に対する異常検知器開発を踏まえ、XRISM 衛星に関する異常検知器開発を行う。XRISM 衛星の異常検知器を開発するにあたり、すでに XRISM 衛星に搭載される ATMOS を補完するような、異常検知器の開発が求められる。よって今回は、検出器ノイズの異常や装置の HouseKeeping データの時系列パターンに潜む異常である、以下の二つの異常を検知することを大きな目的とした。

1. ノイズスペクトルの異常検知（4.2）
2. 50 mK ステージの異常温度変化検出 (4.3)

これら二つの異常は、XRISM 衛星に関する異常の中でも、*Resolve* 装置という XRISM 衛星に搭載されている極低温検出器に関する異常である。これらの異常について以下で説明する。

まずノイズスペクトルの異常検知について説明する。*Resolve* 装置のエネルギー分解能を上げるためには、スペクトルの安定化が必要がある。しかし観測装置の異常が最も顕著に表れるのは、最も感度が高い検出器であり、その異常をもっともよく表現するのが、*Resolve* 装置の観測中に得られるノイズスペクトルである。ゆえにスペクトルの安定化のために、異常なノイズスペクトルの検知を課題とした。

次に 50 mK ステージの温度異常検知について説明する。*Resolve* 装置の検出器は、通常高エネルギー分解能保持

のため 50 mK に保たれている。しかし *Resolve* 装置に異常が発生た場合や、ADR Recycle の動作が行われると、50 mK から温度が外れる。ADR Recycle とは 50 mK ステージを冷却する断熱消磁冷凍機 (ADR) が、冷却を継続するため、一定時間毎に行う磁化作業を指し、このとき温度は 50mk から変動する。今回は ADR Recycle の開始時刻をはじめとする、50 mK ステージが異常な温度変化をした時刻を、検出することを課題とした。加えて温度変化が、ADR Recycle が原因か、もしくはそれ以外の異常が生じたのかを分類することも課題とした。

そしてこれら目標に対し、今回は閾値によらない検出を目標とするため、従来の方法とは異なる異常検知アルゴリズムを用いる必要がある。そこで、今回は機械学習的手法を中心として、異常検知アルゴリズムを構築した。本論文では、上記の異常を検知するために、機械学習的手法が有用であることを示すことを目標とする。

1.3　構成

異常検知器の開発にあたり、必要となるのが異常を検知するデータへの理解、対象となる異常に適した異常検知アルゴリズムの検討、そして最良なアルゴリズムを選択するための適切な評価である。

これを踏まえ、本論文の構成は次のように構成する。まず §2 で、XRISM 衛星搭載 *Resolve* 装置及びそのデータについて述べ、データ解析のための事前知識を記す。次に §3 では、異常検知アルゴリズムの特性や数学的な背景、異常検知アルゴリズムに対する評価方法を記す。最後に §4 では、実際に検知対象の異常に対して異常検知器の開発を行い、その結果を記す。

§4 では主に三つの課題に取り組んだ。初めに §4.1 では、ノイズスペクトルの類別を行う。これは本論文の直接の目的でないが、検出器ノイズに対し機械学習的手法が利用できるかを確認した。次に §4.1 での手法を踏まえ、4.2 では、ノイズスペクトルの異常検出に取り組んだ。最後に §4.3 では、50 mK ステージの温度異常検知に取り組んだ。

第 2 章

XRISM 衛星搭載 *Resolve* の装置とデータ

Contents

2.1 装置

2.1.1 XRISM 衛星

XRISM 衛星とは、NASA や ESA との国際協力により、2018 年に開始された、JAXA 宇宙科学研究所の 7 番目の X 線天文衛星である。XRISM 衛星は 2016 年に運用を停止した X 線天文衛星 ASTRO-H 衛星の後継機であり、ASTRO-H 衛星が担っていた超高分解能 X 線分光による、宇宙物理学の課題の早期かつ確実な遂行が目標となる。図（2.1）は XRISM 衛星の模式図である。

XRISM 衛星の取り組む課題は大きく分けて三つある。具体的には、（1）銀河団のダイナミクスの解明、（2）宇宙にある元素の生成方法の解明、（3）コンパクト天体周りのプラズマ構造の解明である。

初めに（1）に関して説明する。まず銀河団とは、明るい銀河を 100 個程度以上含む銀河の集団である。銀河団は基本的に、銀河団中の暗黒物質による重力場と高温プラズマ圧力のバランスで成り立っている。しかし、高温プラズマはX線放射をしながら次第に冷め、高温プラズマの温度は次第に下がることが予想される。温度が下がることで、圧力も下がり銀河中心部の密度が高まる。そして密度が高まることで、放射効率が上がり、温度がさらに下がる。よって密度の高い中心部は、一億年から十億年程度で、冷却してしまうはずである。しかし実際には、銀河団の高温ガスは百億年のスケールで安定しており、何が X 線放射によるエネルギー流出を補填するのかが、大きな疑問点となる。このとき、エネルギー流出の補填を行うとされる候補が三つある。一つ目が、周辺の冷却されていないプラズマからの熱伝導である。二つ目が、高温プラズマ中で運動する銀河からのエネルギー供給である。三つ目が中央部の大きな銀河の超巨大質量ブラックホールからのプラズマ流の過熱である。これら三つの候補のうちどれが正しいかを、検証するのが（1）の課題である。それぞれの候補に対し、一つ目の候補に対しては、高温プラズマの温度分布の精密な測定によって、二つ目、三つ目の候補に対しては、銀河周辺の高温プラズマ運動の様子の測定によって明らかにする。

次に（2）に関して説明する。銀河団中の高温プラズマを解析することで、宇宙史的な規模で元素合成の歴史が

見える。恒星や超新星で作られた金属元素は、水素を主とする銀河間プラズマのなかで豊饒さを増していく。恒星や超新星の種類によって、作られる元素の組成パターンは異なるので、それぞれの組成パターンを測定することで、数十億年にわたる元素合成の歴史や、それを生み出した恒星、超新星の歴史を知ることができる。XRISM 衛星は、X 線分光によって、微量の元素の割合や拡散を測定できる。これにより元素合成の知見が、より明らかになる。

最後に（3）について説明する。コンパクト天体とは、ブラックホールや中性子性、白色矮星といった恒星が、終わりを迎えた後に残される天体である。コンパクト天体は、周囲に強い重力場をもち、それに引き寄せられた渦巻くプラズマを伴う。このプラズマを降着円盤もしくは降着流と呼ぶ。降着円盤は一般相対論が支配する重力場における、時空法則を観測するために、重要な手掛かりとなる。ゆえに降着円盤を調べることは、時空構造の研究に対し重要な意味を持つ。そのために降着円盤からの X 線スペクトルを超高分解能 X 線分光によって仕分け、観測する必要がある。

以上のような課題を究明するため、XRISM 衛星は 2 つのミッション機器を搭載している。1 つ目が *Resolve* 装置とよばれる、超高分解能でエネルギーを図るX線マイクロカロリメータである。2 つめが Xtend とよばれる、広い波長域で画像を取る X 線 CCD カメラである。それぞれのミッション機器には、XMA とよばれる、透過力の高い X 線を集める X 線望遠鏡が付随している。これらのミッション機器の中で、*Resolve* 装置は、日本、アメリカ、オランダ、スイスによりで開発され、次世代の X 線天文学を担う新しい観測装置として期待されている。

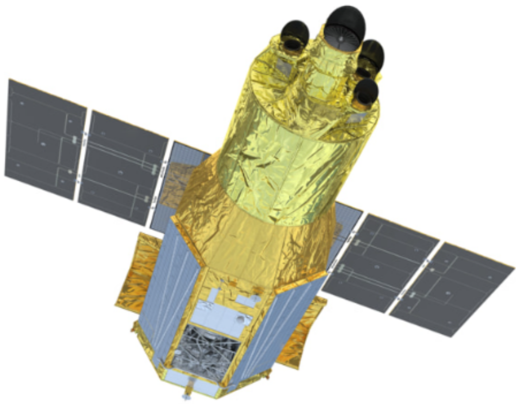

図 2.1. XRISM 衛星の模型図 (2)

2.1.2　*Resolve* 装置

検出器

■X線マイクロカロリメータ　*Resolve* 装置はX線マイクロカロリメータを用いて、入射したX線のエネルギーを光子ごとに測定する。(11)*Resolve* 装置のX線マイクロカロリメータは、一辺 $814\mu m$ の正方形であり、6×6 のピクセルで構成される。図（2.2）が *Resolve* 装置におけるX線マイクロカロリメータのピクセルの配置図である。以下ではX線マイクロカロリメータの原理やエネルギー分解能について説明する。

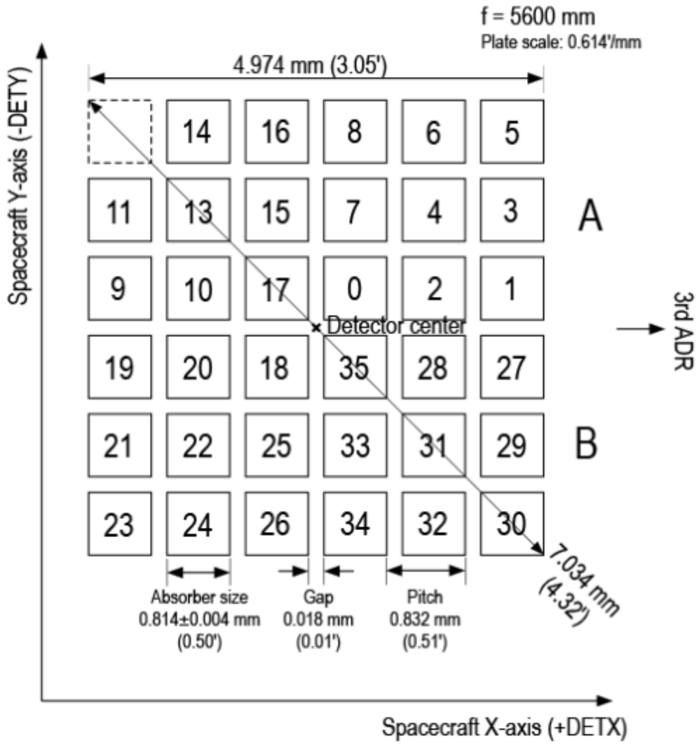

図 2.2. *Resolve* 装置のピクセル図 (3)

受光素子が X 線を吸収すると、X 線のエネルギーは熱に変わる。よって、X 線を吸収した熱により、X線マイクロカロリメータの受光素子は、温度が上昇する。その上昇した温度を測定することで、入射した X 線のエネルギーを測定するのが、X線マイクロカロリメータの基本的な原理である。例えば、入射した X 線光子のエネルギーを E、受光素子の熱容量を C とすると、受光素子の温度変化 ΔT は

$$\Delta T = \frac{E}{C} \tag{2.1.2.1}$$

となる。この ΔT は微小であり、温度の微小変化を測定するためには、極低温に冷やす必要がある。X線マイクロカロリメータの原理を、図示したものが図（2.3）である。

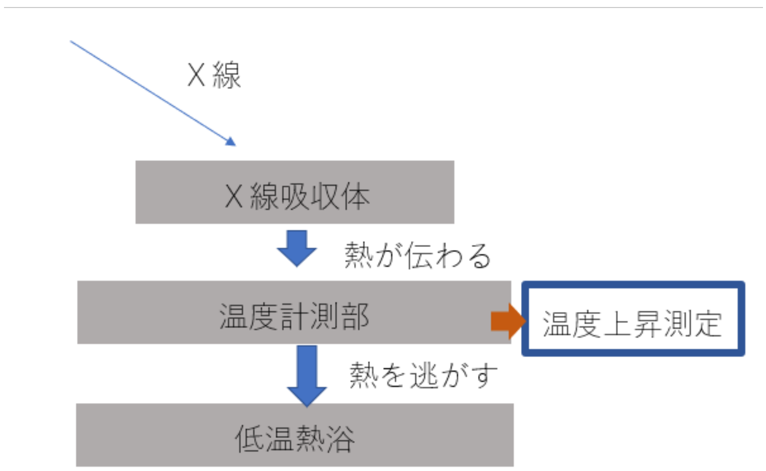

図 2.3. X線マイクロカロリメータの作動原理

次にX線マイクロカロリメータの、エネルギー分解能について説明する。まずX線マイクロカロリメータに入射した X 線のエネルギーは、スペクトルデータから導き出す。ゆえに、エネルギー分解能を上げるためには、スペクトルの半値全幅を小さくする必要がある。そこで温度 T の物質中で、X 線が入射した時の、スペクトルの半値全幅を求める。温度 T のとき、X 線光子一個が持つ平均エネルギーは、ボルツマン定数 k_B とすると、

$$\epsilon = k_B T \tag{2.1.2.2}$$

となる。またX線マイクロカロリメータの受光素子の熱容量 C をもちいると、受光素子の内部エネルギーは

$$U = CT \tag{2.1.2.3}$$

と表される。これより平均 X 線光子数は、

$$N = \frac{CT}{k_B T} = \frac{C}{k_B} \tag{2.1.2.4}$$

と導かれる。光子数はポアソン分布に従うので、光子数の分散は $\sqrt{N}$ となる。よって、受光素子の内部エネルギーの分散は、

$$\Delta U \simeq \Delta N k_B T = \sqrt{k_B C T^2} \tag{2.1.2.5}$$

となる。これをフーリエ変換したときの半値全幅 FWHM は

$$\mathrm{FWHM} \simeq 2.35\sqrt{k_B C T^2} \tag{2.1.2.6}$$

となる。(A.1) しかし実際には温度計の感度などが関係してくるので、それを考慮した係数 ϵ をもちいると

$$\mathrm{FWHM} \simeq 2.35\epsilon\sqrt{k_B C T^2} \tag{2.1.2.7}$$

となる。式 (2.1.2.7) より、エネルギー分解能を向上させるためには、温度 T と熱容量 C を小さくする必要がある。しかし設計が決まった場合、温度と熱容量は変えることができない。ゆえに、X 線マイクロカロリメータのエネルギー分解能を高めるには、設計段階で温度と熱容量を小さくすると同時に、設計時の分解能を再現するために、ノイズスペクトルの影響を小さくする必要がある。

■反同時計測検出器　反同時計数検出器 (anti-coincidence detector,Anti-co 検出器) は、宇宙線などの荷電粒子のバックグラウンド除去のための検出器である。大きさは 10 mm×10 mm で、厚さは 500 μm である。Anti-co 検出器の模式図は図 (2.4) である。Anti-co 検出器はX線マイクロカロリメータの下に設置され、信号をX線マイクロカロリメータと Anti-co 検出器で同時に検出した時、その信号を除去する。これにより。マイクロカロリメータを貫通する宇宙線起源のバックグラウンドを除去できる。

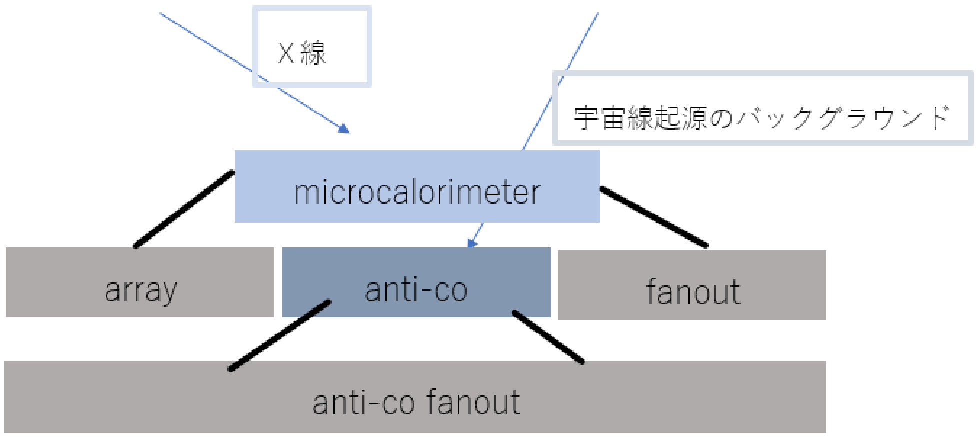

図 2.4. 反同時計測検出器の構造

信号処理系

■X-ray Box　X-ray Box (XBOX) は *Resolve* 装置のアナログ信号処理部である。[cite]XBOX はX線マイクロカロリメータと反同時計測検出器からの信号を読み出し、信号のフィルタリング、増幅と A/D 変換を行う。XBOX には、A 系統、B 系統の二つの系統が存在し、それぞれがカロリメータアレイの 18ch と、反同時計数検出器の 1ch の信号を扱う。このとき反同時計数検出器は一つしかないが、その信号は分岐され A/B 両系で読みだされる。A/D 変換されたデジタルデータは、Low Voltage Differential Sinaling (LVDS) により、デジタル波形処理部の対応する系である Pulse Shape Processor（PSP）に送られる。ここで LVDS とは低電圧差動信号のことを指し、差動より信号を送るのでノイズに強く、また低電圧で送ることにより低消費かつ高速で伝送できるのが特徴である。XBOX はデータ処理のほかに、カロリメータや反同時計数検出器へのバイアス電圧供給や増幅器の温度制御、さらにその電圧と温度のモニタリングを行う。

■Pulse Shape Processor　Pulse Shape Processor (PSP) は *Resolve* 装置のデジタル波形処理部である [cite]。PSP には XBOX 同様、冗長系のため、機能的に同等な A 系、B 系の二つの系統が存在する。それぞれに対応する系の XBOX から、デジタル変換されたデータを受け取り、微分波形を用いたパルスの検出、イベントのグレード付け、最適フィルタ処理による光子のエネルギー及び到来時刻の高精度測定、ノイズ収集を行う。以下では各作業について説明する。

まずパルスの検出について、説明する。複数光子を短時間に検出するとき、初めに到達したイベントをファーストパルス、二番目以降に到達したイベントをセカンドパルスという。ファーストパルス検出は FPGA によって、セカンドパルス検出は CPU によって行われる。検出方法は、まず FPGA で、ADC sample についての時間微分が計算される。ADC sample とは、ある時刻の個々のピクセルの波高値データのことを指す。ADC sample は XBOX から 12.5 kHz で送られてくるので、ADC sample の時間間隔は、8×10^{-5} の秒となる。これらの状況をもとに微分値を計算し、微分値が閾値を超えた時に、ファーストパルスにトリガーをかける。その後、CPU によって、ファーストパルスのイベント波形から平均波形を引き、その差分がセカンドパルス用の閾値以上の時、セカンドパルスがトリガーされる。

次にイベントのグレード付けについて説明する。PSP で評価する波形の中に複数のイベントが重なると、正確なエネルギー計測ができない。そのためイベントの前後の時間間隔によって、イベントは分類される。これをグレード付けという。ここでイベントの到来時間間隔は、ADC sample が送られる時間間隔の 8×10^{-5} 秒の $\frac{1}{16}$ を単位とする。まず、High Resolution (HR) イベントは、イベントがトリガされた時間の前後 874 ADC sample の間 (69.92 ms) に、ほかのイベントが到来していないイベントを指す。Medium Resolution(MR) イベントは、イベントがトリガされた時間から、256 ADC sample(17.52 ms) の間に、ほかのイベントが入射していないイベントを指す。イベントがトリガされてから、ほかのイベントが入射するまでの時間がそれ以下のイベントは、Low Resolution(LR) イベントにグレード付けされる。次に、primary、secondary については、その前のイベントからの関係を考え、先に来たものを primary、次に来たものを secondary とする。具体的には、前のイベントから十分離れているイベントは primary、離れていないイベントを secondary とグレード付けする。LR イベント、MR イベント、HR イベントの順かつ、secondary、primary の順でよいエネルギー分解能が得られる。これは前後の影響が少ないほうが、波形のゆがみが少なくなるので、下述する最適フィルタ処理でエネルギーや光子の到来時間を正確に処理できるからである。図（2.5）はそれぞれのイベントの例を表したものである。

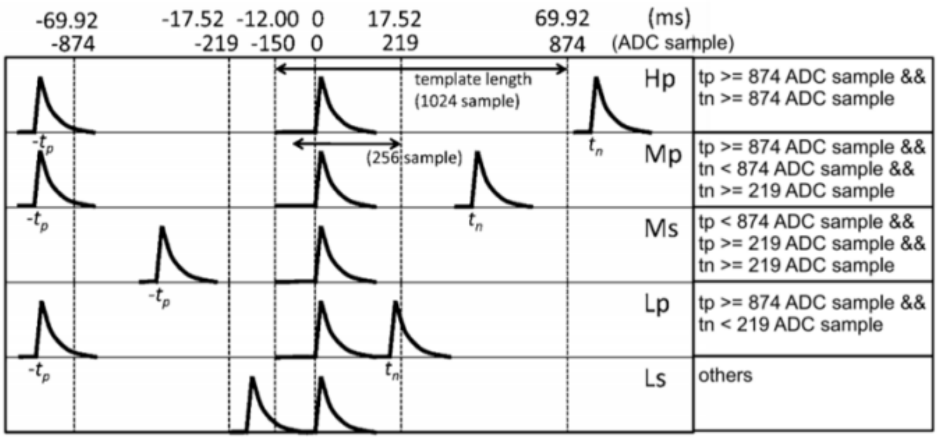

図 2.5. イベントのグレード付け (4)

最後に最適フィルタ処理について説明する。PSP での波高値解析は、最適フィルタ処理によって行われる。最適フィルタ処理とは、ノイズを含む実データから最適な波形パラメータを求める処理である。(site) これにより、入射した光子のエネルギーと到来時刻を高精度で測定することができる。以下では最適フィルタ処理のアルゴリズムを説明する。

まず X 線が入射した時、時定数は X 線のエネルギーによらないため、すべてのパルスは同じ形をしていると仮定する。このとき、実データ $D(t)$ とし、規格化されたパルス波形を $S(t)$ とすると、

$$D(t) = HS(t) + N(t) \tag{2.1.2.8}$$

のようにあらわされる。この時の $N(t)$ がノイズデータであり、H がパルス波形の大きさである。これを周波数空間において考え、規格化された実データ、パルス、ノイズをフーリエ変換したものをそれぞれ、$D(\omega)$、$S(\omega)$、$N(\omega)$ とすると、フーリエ変換の線形性より

$$D(\omega) = HS(\omega) + N(\omega) \tag{2.1.2.9}$$

となる。この時最適な H とは、実データとパルス波形との差を最小にするものなので、

$$\chi^2 = \sum_{\omega} \frac{|D(\omega) - HS(\omega)|^2}{|N(\omega)|} \tag{2.1.2.10}$$

を最小とする H を求めればよい。よって

$$\frac{\partial}{\partial H}\chi^2 = 0 \tag{2.1.2.11}$$

を求めると

$$\begin{aligned} H &= \frac{\sum_\omega \frac{D(\omega)S^*(\omega)+D^*(\omega)S(\omega)}{2|N(\omega)|^2}}{\left|\frac{S(\omega)}{N(\omega)}\right|^2} \\ &= \frac{\sum_\omega \frac{D(\omega)S^*(\omega)}{|N(\omega)|^2}}{\left|\frac{S(\omega)}{N(\omega)}\right|^2} \\ &= \frac{\sum_\omega \frac{D(\omega)}{S(\omega)}|\frac{S(\omega)}{N(\omega)}|^2}{\left|\frac{S(\omega)}{N(\omega)}\right|^2} \end{aligned} \tag{2.1.2.12}$$

と求まる。式 (2.1.2.12) より、これは $\left|\frac{S(\omega)}{N(\omega)}\right|^2$ を重みとしたときの、$\frac{D(\omega)}{S(\omega)}$ の平均値を表していることがわかる。よって、逆フーリエ変換を行うと

$$H = \sum_t \frac{\frac{D(t)}{S(t)}|\frac{S(t)}{N(t)}|^2}{\sum_t |\frac{S(t)}{N(t)}|^2} \tag{2.1.2.13}$$

と求まる。これが最適フィルタ処理である。

冷却系

Resolve 装置の冷却系は、冷媒 (超流動ヘリウム)、一台の ^{4}He Joule-Thomson 冷凍機 (JTC)、四台の 2 段 Stirling 冷凍機 (STC)、三台の断熱消磁冷凍機 (ADR) からなる。冷却チェーンは図 2.6 のようになっており、X線マイクロカロリメータは、Dewar(DWR) というクライオスタットの中に保持されている。DWR 内で複数台の機械式冷凍機により 4 K まで冷やされ、超流動ヘリウムによって 1.2 K 程度まで冷やされる。1.2 K 以下は二段の ADR によって、50 mK まで冷却される。

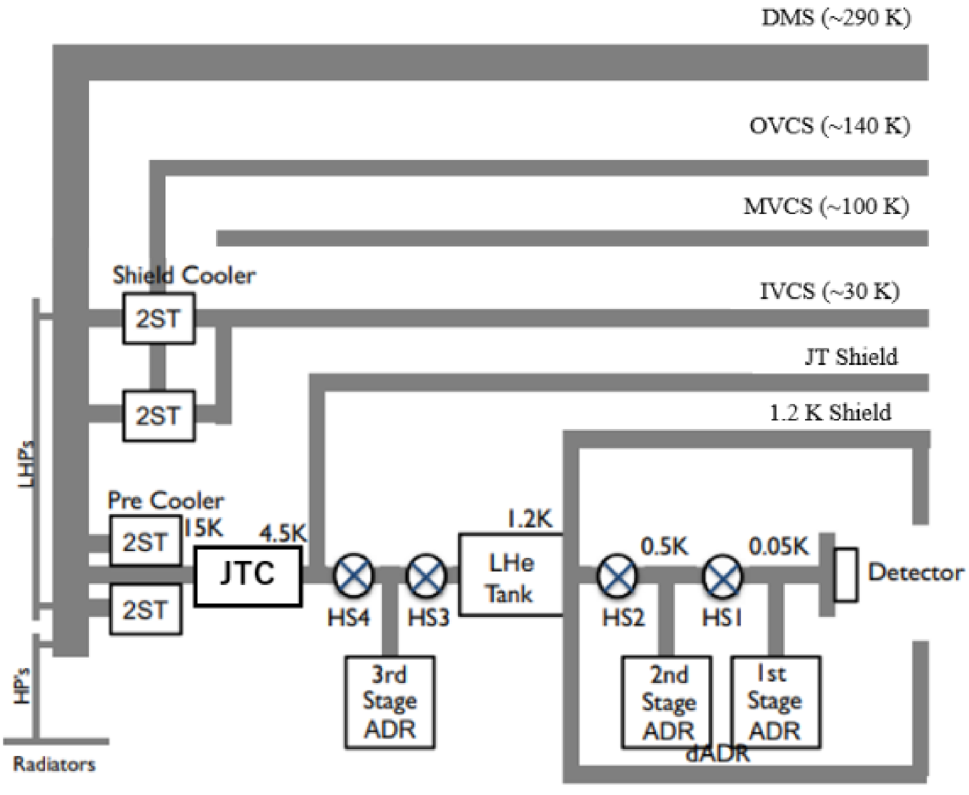

図 2.6. *Resolve* 装置の冷却チェーン (3)

■DWR　DWR とは、検出器を内部に保持するクライオスタットである。クライオスタットとは、真空低温槽のことを指し、*Resolve* 装置では、内外に保持するほかの冷凍機と合わせて、検出器を 50 mK に冷却する。ゆえに、DWR 内には複数の冷凍機と検出器が保持されている。図 2.7 が DWR の断面図である。DWR は外側から Dewar Main Shell(DMS)、Outer Vaper-Cooled Shield(OVCS)、Middle Vaper-Cooled Shield(MVCS)、Inner Vaper-Cooled Shield (IVCS)、Joule-Thomson Shield (JTS) で構成されている。これらのシールド内は真空に保

たれており、シールド間には Multi-layer insulator ML1 という断熱材が挟まれ、断熱化されている。それぞれのシールドにより、徐々に室温である約 300 K から、一番内側では検出器を 50 mK まで冷やされていく。

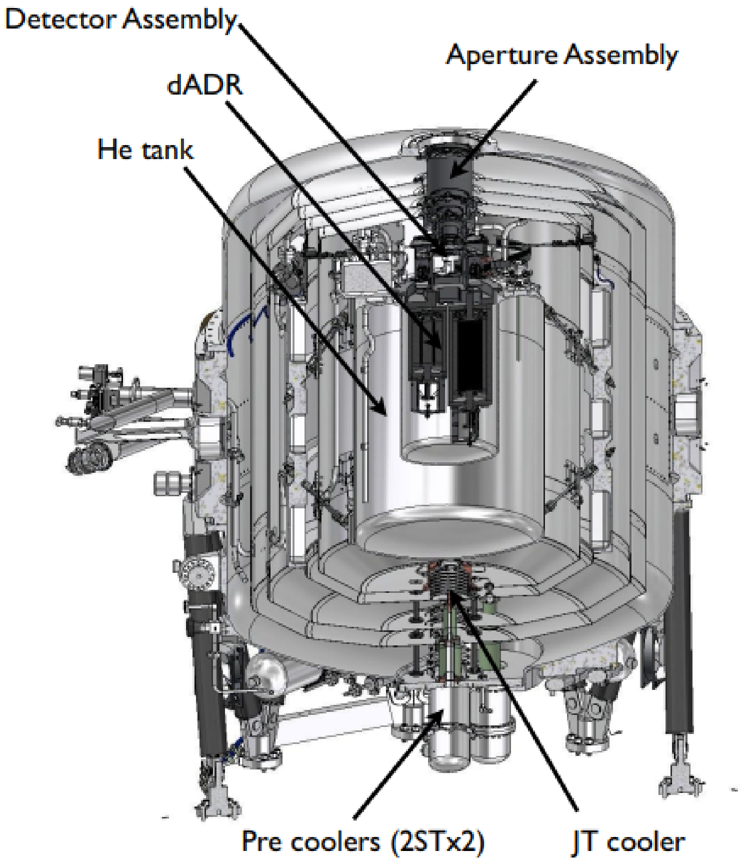

図 2.7. CAD で書かれた DWR の断面図 (3)

■超流動ヘリウム　冷媒を用いる冷却では、固体もしくは液体の冷媒をタンクで保持し、その潜熱および顕熱を利用して冷却する。*Resolve* 装置の冷却系では、冷媒として超流動ヘリウムを使用しており、DWR の一番内側の He タンクに収納される。

超流動ヘリウムは大気圧において、沸点が約 4 K である。さらに、気圧がほぼ 0 の宇宙空間においては、沸点がさらに低くなり、より低温の 1.2 K まで冷却できる。地上ではこの真空状態に近づけるため、排気を行うことで 1.2 K までの冷却を行う。

■断熱消磁冷凍機　断熱消磁冷凍機 (ADR) とは、磁場により常磁性塩の温度-エントロピー曲線 (T-S グラフ) を操作することで冷却する冷凍器である。*Resolve* 装置の冷却系では、ヘリウムタンクと検出器の間にある 1st Stage ADR、2nd Stage ADR、そして Joule-Thomson 冷凍機とヘリウムタンクの間にある 3rd Stage ADR の三つが使用されている。3rd Stage ADR は通常は作動しないが、超流動ヘリウムがなくなった後に作動する。これにより、超流動ヘリウムがなくなった後も、ヘリウムタンクを冷却することが可能となる。加えて、3rd Stage ADR は、ヘリウムタンクから、熱を吸い出すことができるため、通常観測時から用いた場合、超流動ヘリウムの寿命を延ばすことができる。

ADR は磁性塩への磁場を調節し、断熱消磁を利用して冷却する。ここで、断熱消磁の原理について、N 個のスピンの系を用いて説明する。(12) ただし今回は問題を簡単にするため、スピン間に相互作用がない系を考える。温

度 T、磁場 H で熱平衡状態の場合、磁気モーメントを $\vec{\mu}$、スピンを $\vec{s}$、とすると、

$$\vec{\mu} = \mu_0 \frac{\vec{s}}{\hbar} \tag{2.1.2.14}$$

となる。そこで、スピンの大きさを $\sigma\hbar$ とすると、一つのスピンのエネルギー E は、

$$E = -\mu H = -\mu_0 H\sigma \tag{2.1.2.15}$$

と表される。よって $\beta = k_B T$ とすると、この系での分配関数 Z は、

$$\begin{aligned} Z &= \left(\sum_{\sigma_i=-1,1} exp(-E_i\beta) \right)^N \\ &= (\exp\beta\mu_0 H + \exp -\beta\mu_0 H)^N \\ &= (2\mathrm{cosh}(\beta\mu_0))^N \end{aligned} \tag{2.1.2.16}$$

となる。このとき分配関数 Z と系のエントロピー S は、ヘルムホルツの自由エネルギー F を用いて、

$$S = -\left(\frac{\partial F}{\partial T}\right)_V = \left(\frac{1}{\partial T}\left(\frac{\mathrm{log}Z}{\beta}\right)\right)_V \tag{2.1.2.17}$$

のような関係が成立する。よって、この系のエントロピー S は

$$S = Nk_B \log\left(2\cosh\frac{\mu_0 H}{kT}\right) - \frac{N\mu_0 H}{T}\tanh\frac{\mu_0}{k_B T} \tag{2.1.2.18}$$

と求まる。式 (2.1.2.18) を見ると、系におけるエントロピーは H/T の関数となっている。これより、エントロピーを一定に保ちながら外部磁場を減少させると、H/T が一定より温度が下がることがわかる。これが断熱消磁の原理である。ただしこのとき、磁場 H と温度 T が比例関係にあることから、$H = 0$ のとき、$T = 0$ となるはずだが、実際はスピン間の相互作用の影響があり、エントロピーは低温で、式 (2.1.2.18) からずれる。よって $H = 0$ でも、温度が 0 K にとなることはない。

図 (2.8) は ADR の冷凍サイクルの S-T グラフの概要図である。(a) は等温で磁場をかけて、磁場を $H = 0$ から $H = H' > 0$ に増やしている過程である。この際に熱が発生するが、その熱はヘリウムタンクに捨てられる。(b) はそこから磁場を下げ、磁場を $H = H'$ から $H = 0$ にし、断熱消磁を行っている。(a)、(b) のサイクルを繰り返し、冷却を行うのが、ADR の仕組みである。

Resolve 装置では機械式冷凍機により 4.5 K まで予冷されている。そこから He Tank 温度は $1.2K$ であり、これを起点として、ADR 二つを直列することで、4.5 K から 50 mK まで一回で冷却される。ADR は（b）の断熱消磁による外部磁場の減少が終わると、最大一時間かけて（a）のような再磁化する。これを ADR Recycle という。ADR Recycle 時は、温度が上昇し 50 mK を保持できなくなることから、異常が発生しやすくなる。また 50 mK まで冷却されてからは、50 mK を調節するため、magnet 電流を調節して温度を保持している。

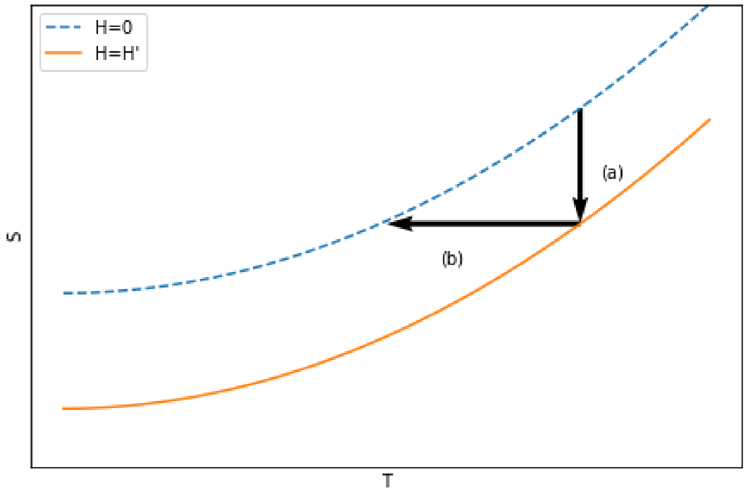

図 2.8. ADR の冷却サイクルの T-S グラフ

■Joule-Thomson 冷凍機　Joule-Thomson 冷凍機 (JTC) はジュールトムソン効果を利用した冷凍機である。*Resolve* 装置では二台の Joule-Thomson 冷凍機を用いて、ヘリウムタンクの放射シールドの一つである、JT Shield を冷却する。

Joule-Thomson 冷凍機の仕組みについて述べるため、ジュールトムソン効果について説明する。(13) そのために図 (2.9) のような断熱壁で囲まれ、細孔栓で仕切られた、左右二つの室にピストンを備えた系を考える。ジュールトムソン効果とは、このような系で両室の圧力を一定に保ちながら、細孔栓を通じ内部気体を移動させたとき、内部気体の温度が変化する効果のことを指す。これについて、式を用いて説明する。

まず両室の圧力を一定 p_1、p_2 に保ちながら、ピストンを押すことで、左側の体積が V_1 から 0、右側の体積が 0 から V_2 となったとする。このとき、外界から加えられた仕事 W は

$$W = -\int_{V_1}^{0} p_1 dV - \int_{0}^{V_2} p_2 dV = p_1V_1 - p_2V_2 \tag{2.1.2.19}$$

となる。次に、熱力学第一法則より、内部エネルギーの変化量は、ピストンを動かす前の内部エネルギーを U_1、ピストンを動かした後の内部エネルギーを U_2 とすると

$$U_2 - U_1 = p_1V_1 - p_2V_2 \tag{2.1.2.20}$$

となる。よって、

$$U_1 + p_1V_1 = U_2 + p_2V_2 \tag{2.1.2.21}$$

となる。このときエンタルピー H は、

$$H = U + pV \tag{2.1.2.22}$$

であるので、ピストンを押す前後で、気体のエンタルピーが保存している。つまりこの系で両室の圧力を保ちながら、細孔栓を通じ内部気体を移動させたとき、気体のエンタルピーは保存される。

次に、エンタルピー H の微小変化量は式 (2.1.2.22) を用いると、

$$dH = dU + d(PV) = dQ + Vdp = TdS + Vdp = T\left(\frac{\partial S}{\partial T}\right)_p dT + T\left(\frac{\partial S}{\partial p}\right)_T dp + Vdp \tag{2.1.2.23}$$

と表される。この式を変形する。まず定圧比熱の定義より、

$$C_p = \frac{\partial Q}{\partial T} = T\left(\frac{\partial S}{\partial T}\right)_p \tag{2.1.2.24}$$

となる。次に、またギブスの自由エネルギー G は

$$G = U + PV - ST \tag{2.1.2.25}$$

であることから、ギブスの自由エネルギーの微小変化量は

$$dG = Vdp - SdT \tag{2.1.2.26}$$

となる。ここから

$$\left(\frac{1}{\partial T}\left(\frac{\partial G}{\partial p}\right)_T\right)_p = \left(\frac{1}{\partial p}\left(\frac{\partial G}{\partial T}\right)_p\right)_T \tag{2.1.2.27}$$

となるので式（2.1.2.26）から

$$\left(\frac{\partial V}{\partial T}\right)_p = -\left(\frac{\partial S}{\partial p}\right)_T \tag{2.1.2.28}$$

が成立する。これより、式 (2.1.2.24) と、式 (2.1.2.28) を式 (2.1.2.23) に代入すると、

$$\delta H = C_p dT + \left[V - T\left(\frac{\partial V}{\partial T}\right)_p\right] dp \tag{2.1.2.29}$$

のようにエンタルピーの微小変化量が表される。

この式 (2.1.2.29) を用いて、先ほどの系でピストンを押したときの、温度変化を考える。先ほどの系では、エンタルピーの値がピストンを移動させたの前後で、変化しないことより、$\delta H = 0$ でとなるので、

$$C_p dT + \left[V - T\left(\frac{\partial V}{\partial T}\right)_p\right] dp = 0 \tag{2.1.2.30}$$

となる。これを変形すると

$$\mu = \left(\frac{\partial T}{\partial p}\right)_H = \frac{[T(dV/dT)_p - V]}{C_p} \tag{2.1.2.31}$$

となる。このときの μ は、エンタルピー一定で気体の圧力を変化させたときの、温度変化量を表し、ジュールトムソン係数といわれる。つまりこの値が、この系でのピストンの変化による温度変化量である。ジュールトムソン係数は、気体のモル数が 1 mol のとき、ファン・デル・ワールスの式より、

$$\left(p + \frac{a}{V^2}\right)(V - b) = RT \tag{2.1.2.32}$$

となるので、p を一定として、T で微分すると

$$0 = \frac{R}{V - b} - \frac{RT}{(V - b)^2}\left(\frac{\partial V}{\partial T}\right)_p + \frac{2a}{V^3}\left(\frac{\partial V}{\partial T}\right)_p \tag{2.1.2.33}$$

となる。ゆえに

$$\left(\frac{\partial V}{\partial T}\right)_p = \frac{R(V - b)V^3}{RTV^3 - 2a(V - b)^2} = \frac{(V - b)/T}{1 - 2a(V - b)^2/RTV^3} \tag{2.1.2.34}$$

となる。よって、$b/V \ll 1$、$2a/RTV \ll 1$ として、

$$\left(\frac{\partial V}{\partial T}\right)_p = \frac{V}{T} + \frac{1}{T}\left(\frac{2a}{RT} - b\right) \tag{2.1.2.35}$$

となることから、実在気体におけるジュールトムソン係数は

$$\mu = \left(\frac{\partial T}{\partial p}\right)_H = \frac{[T(dV/dT)_p - V]}{C_p} = \frac{1}{C_p}\left(\frac{2a}{RT} - b\right) \tag{2.1.2.36}$$

と導かれる。これより式 (2.1.2.36) からもわかる通り、ジュールトムソン係数、つまりエントロピー一定で気体の圧力を変化させたときの温度変化は、気体の種類や圧力、温度によって変化する。特に $\mu = 0$ となる温度を $T = T_c$ とすると、$T > T_c$ の場合は $\mu < 0$ となり、$T < T_c$ の場合は $\mu > 0$ となる。これがジュールトムソン効果である。

よってジュールトムソン効果より $\mu > 0$ のとき、圧力が低下すると、温度が低下するので、ジュールトムソン冷凍機は、T が T_c 以下になるよう保つことで、冷却を行う。

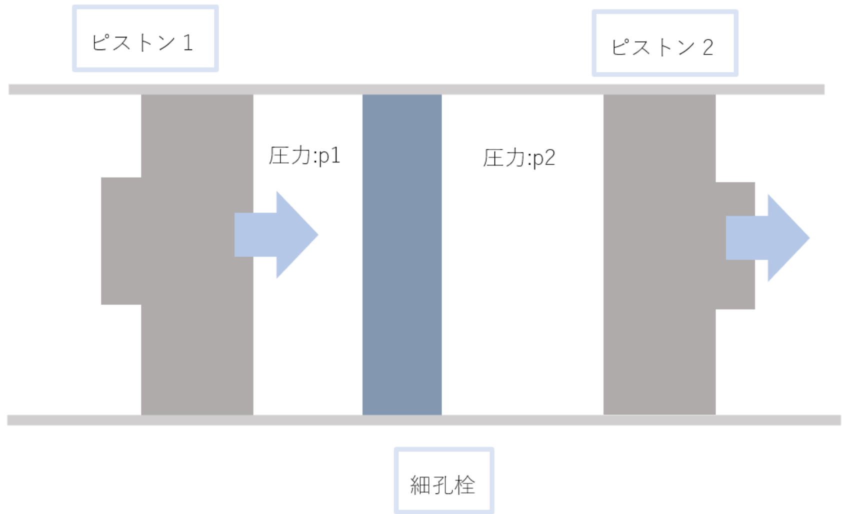

図 2.9. 断熱壁に囲まれ細孔栓で仕切られたピストン

■2 段式 Stirling 冷凍機　2 段式 Stirling 冷凍機 (STC) は、蓄冷器を介した二つのピストンを操作し、冷却を行う。STC のうち二台は Joule-Thomson 冷凍機の予冷に使われ、残る二台の冷凍機は DWR 内にある放射シールドのうち、OVCS、IVCS を冷却するのに使われる。二つの温度ステージを同時に冷却できるので、二段式という。

以下ではスターリング冷凍機の仕組みについて述べる。(14) スターリング冷凍機は蓄冷器を介して、圧縮部と膨張部からなり、以下のようなスターリングサイクルを繰り返すこと冷却される。図 2.8 は以下のサイクルを表したものである。

I. 圧縮ピストンを右に移動する。このとき作動気体が圧縮され、圧縮熱が発生するが、発生した熱はシリンダ周囲に配した冷却水で除去される。
II. 圧縮ピストンと膨張機ピストンが容積を保ったまま右に動かす。これにより、圧縮ガスは蓄冷気を通ることで高圧のまま冷却される。
III. 膨張ピストンを右に動かす。これにより圧縮ガスが膨張仕事を行う。その結果、作動気体の温度は下がり、シリンダ周囲から熱を奪う。
IV. 圧縮ピストンと膨張ピストンを左に動かす。その過程でガスが蓄冷器を冷却し、ガス温度は上昇する。これで 1 サイクルを終了する。

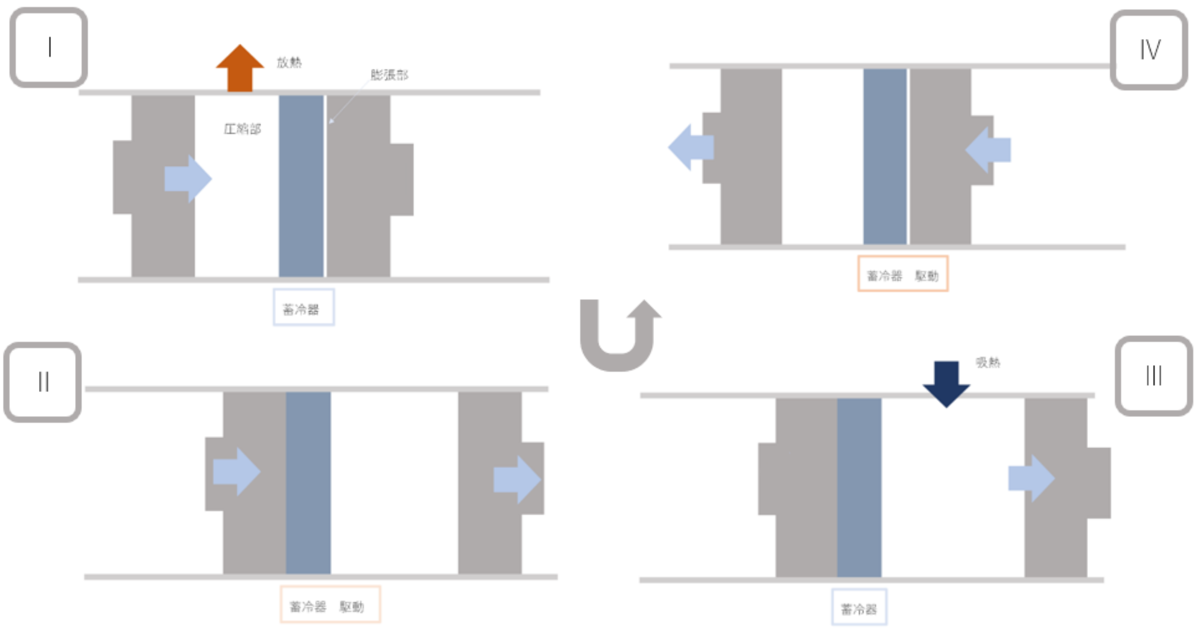

図 2.10. スターリング冷凍機のサイクル

次にこのサイクルの効率について説明する。サイクル I における、圧縮部と膨張部に存在する気体の合計体積を V_1 とする。同様にサイクル II における合計体積を V_2、サイクル III における合計体積を V_3、サイクル IV における合計体積を V_4 とする。加えてサイクル I、サイクル II における気体の温度を T_{12}、サイクル III、サイクル IV における気体の温度を T_{34} とする。このとき、一サイクルの仕事は、1 mol の理想気体の作動ガスにつき、

$$
\begin{aligned}
W &= \oint pdV = RT\oint \frac{dV}{V} \\
&= T_{12}R\log\left(\frac{V_1}{V_2}\right) - T_{34}R\log\left(\frac{V_4}{V_3}\right)
\end{aligned}
\tag{2.1.2.37}
$$

となる。同様に、サイクル III からサイクル IV の間で得た吸収した熱量 Q_{34} を計算すると

$$
Q_{34} = T_{34}R\log\left(\frac{V_4}{V_3}\right) \tag{2.1.2.38}
$$

となる。$V_1 = V_4, V_2 = V_3$ であるので、スターリング冷凍機のサイクルにおける、行った仕事に対する吸収できる熱の比率は

$$
\epsilon = \frac{Q_{34}}{W} = \frac{T_{34}}{T_{12} - T_{34}} \tag{2.1.2.39}
$$

となる。また、スターリングサイクルの P-V 線図を図 (2.11) に示す。

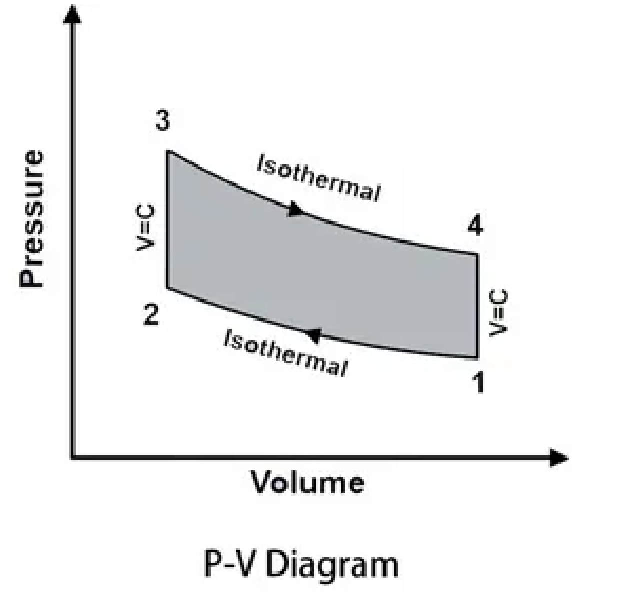

図 2.11. スターリング冷凍機の p-V 図 (左図) とスターリング冷凍機の p-V 図 (右図)(5)

2.2　データ

2.2.1　HK データ

House Keeping データ（HK データ）とは、衛星に搭載されている装置の状態をモニターするデータのことを指す。具体的には装置の温度や消費電流などが HK データに含まれる。本研究では、HK データの中でも、*Resolve* 装置検出器がある 50 mK ステージの温度データに着目する。50 mK の温度データにはモニター用とコントロール用の二種類の温度データがある。モニター用は実際の温度であり、コントロール用とは 50 mK の温度に一定に保つため、PID 制御をする際に用いる温度である。

2.2.2　検出器ノイズデータ

§2.1.2 でも述べた通り、*Resolve* 装置における X 線マイクロカロリメータのノイズデータは、PSP における最適フィルタ処理によって利用される。この節では今回用いた 8k noise spec について説明する。残りのノイズデータについては、§A.1 を参照してほしい。またノイズデータはすべてサンプリング周波数 12.5 kHz で取得している。まずノイズデータについての説明で用いる用語について説明する。

■サンプリング周波数　サンプリング周波数とは、アナログ信号をデジタル信号に変換する際に、一秒間に得る標本の数を指す。*Resolve* 装置の場合は、12.5 kHz がサンプリング周波数となる。

■周波数分解能　周波数分解能 Δf とは、時系列データをフーリエ変換したときの、周波数の最小単位である。これはレコード長が l のとき、

$$\Delta f = 1/l \tag{2.2.2.1}$$

となることから定まる。また、このときの周波数レンジ frange は

$$\mathrm{frange} \geq \Delta f = 1/l \tag{2.2.2.2}$$

となる。またここで最大値は、ナイキスト周波数 $\mathrm{f_{nyg}}$ で決まり、

$$\mathrm{frange} \geq \mathrm{f_{samp}}/2 = \mathrm{f_{nyg}} \tag{2.2.2.3}$$

となる。ここでナイキスト周波数とは、あるサンプリング周波数に対して、この元のアナログ信号に戻すことができる最大の周波数をいう。そしてナイキスト周波数はサンプリング周波数の半分であることが、サンプリング定理より言われている。このナイキスト周波数とは、エイリアスの問題がかかわっている。エイリアスとは、アナログ信号をサンプリングした際に生じる、本来のアナログ信号には存在しない偽の信号のことを指します。例えば図 2.12 のように、4 Hz の波がある場合、この時のサンプリング周波数は 8 Hz 以上である必要がある。例えばサンプリング周波数が 3.5Hz の場合は、元の信号にはない黄色い偽の信号が検出されてしまう。

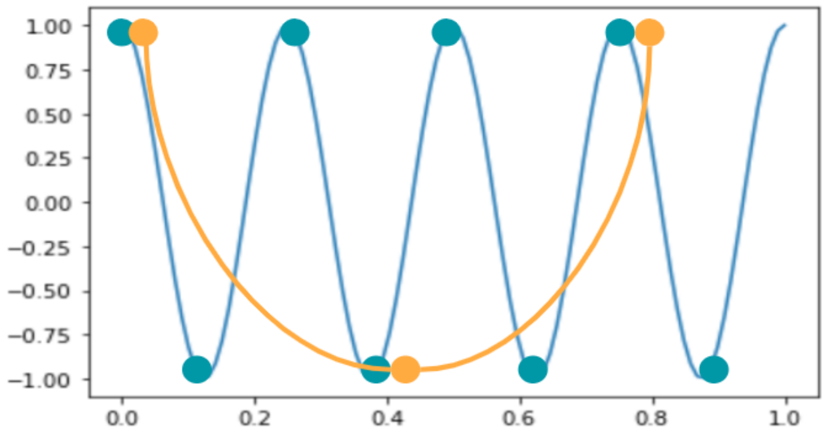

図 2.12. エイリアスについての概要図

8k noise spec

8k noise spec は、周波数帯域が 1.5Hz から 6.25 kHz で取得した、ノイズスペクトルデータである。取得方法は、コマンドで命令したときから、8192 サンプルの長さからなるノイズレコードを 1 セットとして記録する。50 セット、もしくは 10 セットをサンプリングのセット数とする。サンプリングし終えた後、1k noise spec と同様、それぞれをフーリエ変換し、平均化したノイズを取得する。1k noise spec に比べ、レコード長ことから周波数分解能が高いことが特徴である。しかしノイズレコード長が長いため、測定している間に X 線が入射しやすくなるため、取得が一定時間内に完了しないことがある。そこでX線マイクロカロリメータにおける、11、12、13 番のピクセルだけ、サンプリングのセット数を 10 セットとする。これは、12 番目のピクセルに ^{55}Fe の較正用線源を照射しているので、X 線パルスによってノイズレコードが中断されやすいためである。また 11、13 番ピクセルは、12 番ピクセルからのクロストークが発生するため、12 番ピクセルと同様にノイズレコ—ドの取得が中断されやすいためである。

第 3 章

データの異常検知

Contents

3.1 異常検知

3.1.1 非時系列データと時系列データ

非時系列データ

時刻以外の基準において、観測されたデータを並べたものを、非時系列データという。例えば、スペクトルデータは、周波数に対するパワーを並べたものなので、非時系列データといえる。非時系列データは、平均や分散、もしくは複数の説明変数でデータが構成されている場合は相関係数などの統計量を用いて分析される。

非時系列データのなかでも、同一の確率分布に従う確率変数 $X_1, X_2, \ldots, X_n$ がそれぞれ独立であることを independently and identically distributed(i.i.d) という。これはつまり、互いの変数間の共分散が 0 である状態である。機械学習で用いるモデルでは i.i.d が仮定されていることが多い。

時系列データ

時間の経過とともに観測されたデータを、時系列データという。(15),(16) 時系列データのなかには、不規則な間隔で観測されたデータもあるが、以下では一定間隔で観測される時系列データを扱う。時系列データは同じ時系列過程に従う異なる時刻間での自己相関を扱う必要がある。それゆえ時系列データを表す統計量は、平均や分散などに加え、自己共分散・自己相関係数が用いられる。自己相関に関する統計量は、時系列解析において、将来の平均値や変動幅の予測をするのに重要な役割を持つ。例えば、将来の y の期待値や分散の評価を行う際、自己相関が正と分かっている場合は、現在のデータよりも未来の期待値が大きな値になる傾向がわかる。

時系列解析では、観測している時系列データを、ある確率法則を背景に構成されている実現値と理解する。つまり、時系列データ $\{y_t\}_{t=1}^{T}$ を、ある確率変数列 $\{y_t\}_{-\infty}^{\infty}$ からの一つの実現値とみなす。そこで時系列データが従う時系列を確率過程といい、確率過程の構造を表すモデルを、時系列モデルという。時系列モデルの代表的なものとして自己回帰モデル、状態空間モデルなどがある。

時系列データは大きく、定常性を持つ時系列データ（定常時系列データ）と定常性を持たない時系列データ（非定常時系列データ）に分けられる。定常性とはある時系列過程に対し、時間や位置に確率変数が依存せず、一定の確率モデルに従う確率過程であるという性質である。定常性の有無や時系列データの統計量によって、時系列データが従う時系列モデルの選択がされる。

■自己共分散・自己相関係数　期間 $t = 1, 2, \ldots, T$ で観測される時系列データを $Y_1, Y_2, \ldots, Y_T$ とする。このとき Y_t は確率変数となり、平均 $E[Y_t]$、分散 $V[Y_t]$ は

$$E[Y_t] = \mu_t \quad V[Y_t] = \sigma_t^2 \tag{3.1.1.1}$$

と表される。また時系列データにおいて、同一データの異なる時点の共分散を自己共分散 $Cov[Y_t, Y_{t-h}]$、相関係数を自己相関係数 $\rho[Y_t, Y_{t-h}]$ とよび、

$$Cov[Y_t, Y_{t-h}] = \gamma_{t,h} \quad \rho[Y_t, Y_{t-h}] = \frac{Cov[Y_t, Y_{t-h}]}{\sqrt{V[Y_t]V[Y_{t-h}]}} = \rho_{t,h} \tag{3.1.1.2}$$

と表す。

■定常性　定常性は、時間や位置によってその確率分布が変化しない確率過程であるという性質である。定常性である時系列過程は、定常性が成り立つ強さによって、弱定常過程と強定常過程に分けられる。

まず弱定常過程とは、同じ時間差に対して平均と自己共分散が、一定である過程である。具体的には、時系列データの平均と分散が有限の場合かつ、平均と自己共分散が観測時間 t には依存せずに、時間差 h のみに依存する過程を、弱定常過程という。すなわち、Y_t が弱定常過程であるとは、

$$E[Y_t] = \mu \quad Cov[Y_t, Y_{t-h}] = \gamma_{|h|} \qquad (-\infty < \mu, \gamma_{|h|} < \infty) \tag{3.1.1.3}$$

を満たしていることである。

一方強定常過程とは、時間差に対して平均や自己共分散のみではなく、常に同一の分布に従う過程である。具体的には、任意の整数 $h_1, h_2, \ldots, h_n$ に対して、$Y_t, Y_{t-h_1}, Y_{t-h_2} \ldots, Y_{t-h_n}$ の同時分布が時点 t に依存せず、時間差 $h_1, h_2, \ldots, h_n$ にのみ依存する場合、Y_t は強定常過程と呼ばれる。

平均と分散が有限な強定常過程は、弱定常過程である。しかしその逆は成立しない。定常過程の代表的な過程が、iid 系列とホワイトノイズである。

■ホワイトノイズ　共分散定常過程の平均 $\mu = 0$、時間差 $h \neq 0$ のすべての自己共分散 $Cov[Y_t, Y_{t-h}]$ が 0 である場合、その系列をホワイトノイズという。ホワイトノイズは時系列データを用いた回帰分析では、しばしば誤差項が満たすべき過程として想定される。

3.1.2　異常検知アルゴリズム

異常検知アルゴリズムとは、正常なデータと異常なデータが混在するデータセットから、異常なデータを抽出するアルゴリズムである。この時抽出する異常データは図（3.1）のように、四つに大別される。

まず上段のようにルールに従う他の観測値から、値が外れている異常データを外れ値という。特に上段左図のような、時系列のパターンに対して外れている外れ値を時系列外れ値ということもある。これらの外れ値を検知する異常検知を、外れ値検知という。

次に下段のように、他の観測値のふるまうパターンから、ある区間だけ異なるパターンになる地点を検知することを変化点検知という。ただ下段左図のように、外れ値かつ変化点となっている部位を検知することを、変化部位検知ということもある。これらの変化点を検知する異常検知を、変化点検知という。

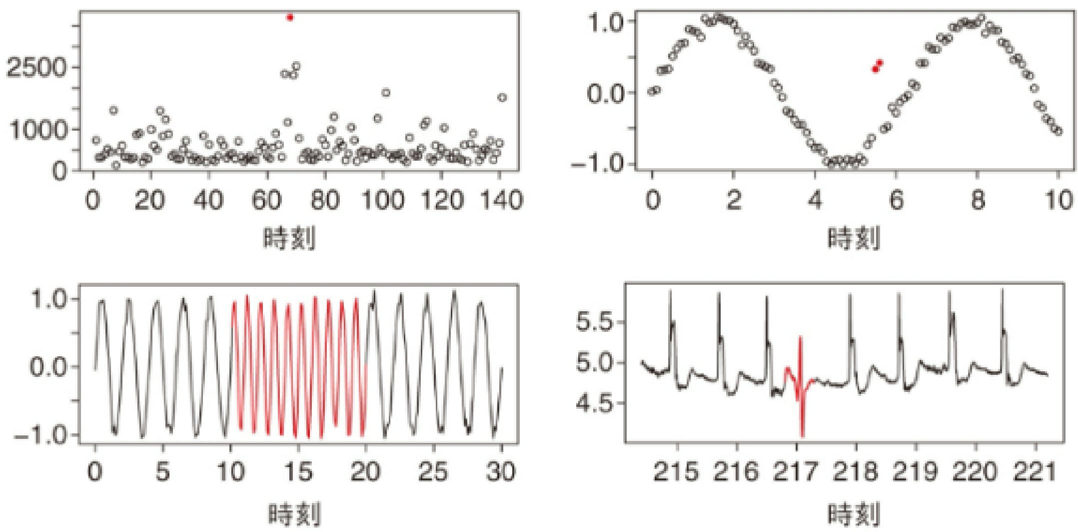

図 3.1. 異常一覧 (6)

外れ値検知

外れ値検知とは、正常なデータが従う確率分布から、大きく外れたデータ点を検知することである。外れ値検出において、異常のデータセットが手に入る場合は、機械学習的手法を用いて、教師あり学習という方法を使い、異常検出器を開発する。教師あり学習とは、正常・異常のラベルがついたデータセットから、ラベル付けの法則を学習し、未知のデータが正常であるか、異常であるかを推測する機械学習的手法の一種である。しかし異常検知を開発する際は、異常データが少ないケースや異常パターンが多種多様なケースが多い。このような場合は、ラベル付のルールを教師あり学習では、学習しきれない。ゆえに正常データセットのみを用いた異常検知アルゴリズムを用いる。このような方法を、半教師師学習を用いた異常検知アルゴリズムという。半教師師学習を用いた異常検知アルゴリズムは図（3.2）のように、大きく四つの手法に分類できる。この中でも今回は、先行研究 (8),(10) でも検討された、再構成誤差 (Reconstruction) を用いて異常検知器を作成した。再構成誤差を用いた異常検知器は、主にクラスタリングを用いた再構成誤差法と、次元削除を用いた再構成誤差法がある。以下ではこの二つの再構成誤差のアルゴリズムを説明する。ただし説明で用いるデータセットについて、正常なデータから構成される訓練データセットを $\boldsymbol{Y_{train}}$、正常と異常なデータが混ざったテストデータセットを $\boldsymbol{Y_{test}}$ とする。それぞれのデータセットは、

$$\boldsymbol{Y_{train}} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{Ntrain}\} \tag{3.1.2.1}$$

$$\boldsymbol{Y_{test}} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{Ntest}\} \tag{3.1.2.2}$$

のような行列で表現されるとする。ここでの $Ntrain$、$Ntest$ はそれぞれ、$\boldsymbol{Y_{train}}$、$\boldsymbol{Y_{test}}$ に含まれるデータ数を表す。そして $\boldsymbol{y}_i (i = 1, 2, \ldots, N)$ は D 次元のデータとする。

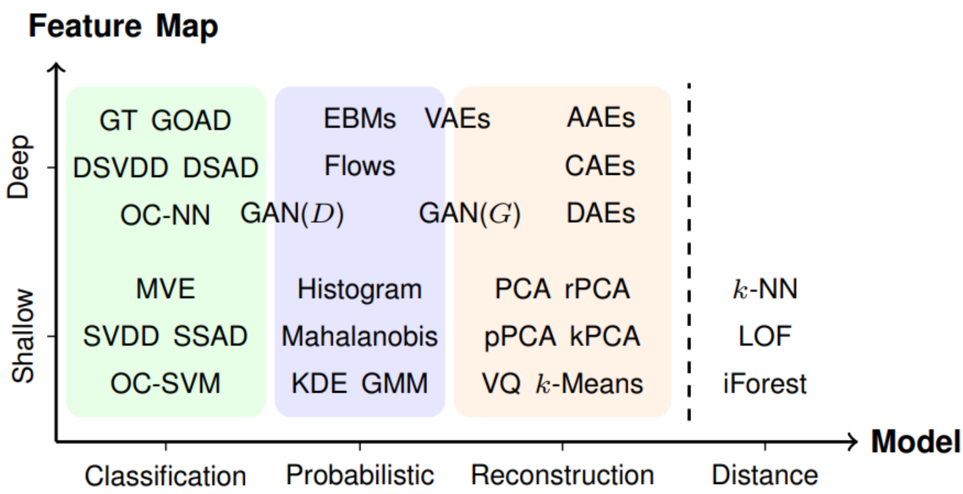

図 3.2. 教師なし学習を用いた外れ値検知アルゴリズム (7)

■クラスタリングによる再構成誤差法　クラスタリングによる再構成誤差法では、正常なデータは一つ以上の特定の形やパターンがあると仮定する。例えば、正常データを測定する環境が常に一定であれば、正常のパターンは一つである。一方環境が時間によって変動する場合、同じ正常データであっても、複数のパターンが形成される。この仮定が成り立つとき、正常データはパターンごとに、D 次元観測空間上で、いくつかのクラスタ（群）を形成すると考えられる。そしてどの正常なクラスタにも属さないデータは、異常なデータと言える。

この仮定をうけ、クラスタリングによる再構成誤差法では、以下のような手順により異常検知を行う。まず正常な訓練データセット $\boldsymbol{Y_{train}} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{Ntrain}\}$ を k 個のクラスタにクラスタリングする。このとき、それぞれのクラスタ中心を $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$ とする。このとき任意のテストデータ $\boldsymbol{y}_i \in Y_{test}$ について、一番近いクラスタ中心が $\boldsymbol{\mu}_i$ であるとき、$\boldsymbol{y}$ に対する再構成ベクトル $\hat{\boldsymbol{y}}$ を

$$\hat{\boldsymbol{y}} = \boldsymbol{\mu}_i \tag{3.1.2.3}$$

とする。このとき元のベクトルと再構成ベクトルとの差である、再構成誤差 $\mathrm{E}_{\mathrm{clustering}}$ は

$$\mathrm{E}_{\mathrm{clustering}} = ||\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i||^2 \tag{3.1.2.4}$$

と表される。異常なデータはどのクラスタにも含まれないので、再構成誤差はある一定の閾値 s よりも大きくなる。故に、テストデータ $\boldsymbol{y}_i$ の再構成誤差 $\mathrm{E}_{\mathrm{clustering}}$ に対して、

$$\mathrm{E}_{\mathrm{clustering}} > s \tag{3.1.2.5}$$

となるとき、そのテストデータ $\boldsymbol{y}$ を異常として検出する。図（3.3）はクラスタリングによる異常検知の概要を図示したものである。

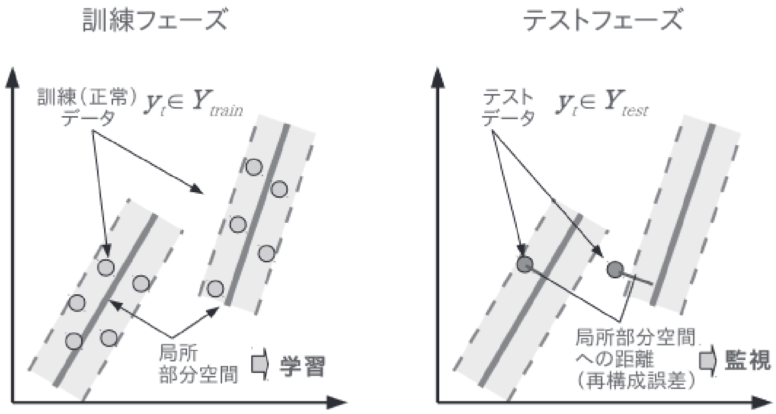

図 3.3. クラスタリングによる異常検知 (8)

■次元削除による再構成誤差　次元削除による再構成誤差では、正常データセットは実データの次元数 D より少ない、次元 d の状態量に縮約できると仮定する。つまり正常なデータセットは、D 次元観測空間の中で、d 次元の部分空間あるいは多様体に拘束されるという仮定である。これは、正常な挙動をする際、データを構成する変量間には一定の相関があり、正常な挙動を特徴づけるパターンが存在すると考えられるからである。この仮定を用いたアルゴリズムでは、最初に正常な訓練データを包括する、部分空間を求める。そこからテストデータと、正常部分空間との距離を測り、一定以上離れたデータを、異常と判定する。

以下では、次元削除による再構成誤差法のアルゴリズムを数式を用いて、具体的に説明する。まず訓練データセット $\boldsymbol{Y}_{train}$ を主成分分析したときに、寄与率が高い d 個の成分を並べた射影行列を $W = [\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_d}]$ とする。このとき、この射影行列によって射影される低次元空間に、任意のテストデータ $\boldsymbol{y}$ を射影したときの座標 $x \in \mathbf{R}^d$ は

$$x = \boldsymbol{W}^T(\boldsymbol{y} - \boldsymbol{\mu}) \tag{3.1.2.6}$$

と表される。ただしこの時の μ は訓練データの平均

$$\boldsymbol{\mu} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \boldsymbol{y}_i \tag{3.1.2.7}$$

である。ゆえにこの時、低次元に射影したものを、実データ空間に再構成したベクトル $\hat{y}$ は

$$\hat{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{\mu} = \boldsymbol{W}(\boldsymbol{W}^T(\boldsymbol{y} - \boldsymbol{\mu})) + \boldsymbol{\mu} \tag{3.1.2.8}$$

とあらわされる。このとき元のベクトルと再構成ベクトルとの差である、再構成誤差 $\mathrm{E_{decomposition}}$ は

$$\mathrm{E_{decomposition}} = ||\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i||^2 \tag{3.1.2.9}$$

となる。正常なデータは射影した部分空間に包括されるので、再構成誤差はある閾値 s 以下となる。一方異常なデータであれば、再構成誤差は閾値より大きくなるので、あるテストデータの再構成誤差 $\mathrm{E_{decomposition}}$ に対して、

$$\mathrm{E_{decomposition}} > s \tag{3.1.2.10}$$

となるテストデータを異常として検出する。図（3.4）は次元削除による異常検知の概要を図示したものである。

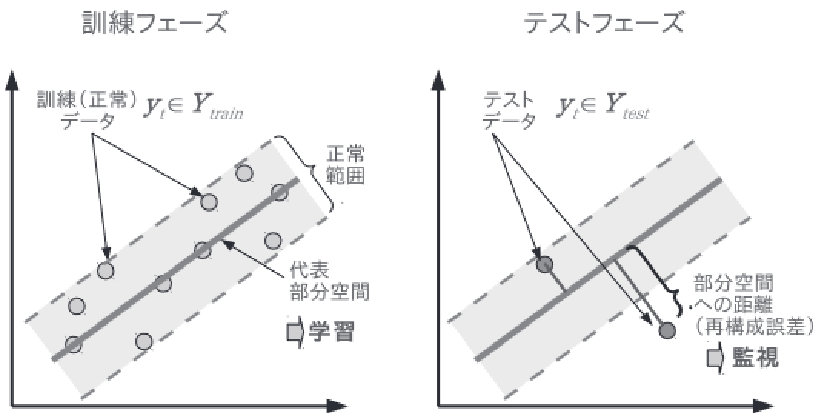

図 3.4. 次元削除による異常検知 (8)

変化点検知

変化点検知とは、連続する時系列データから異常な変化をする時刻を検知することである。(16) 変化点検知をする場合、観測値が事前に決めた閾値を超えた時点で検出するという手法もある。しかし図（3.1）の下段左図のように、閾値をによる手法では検知できない変化点を検知する場合や、閾値を超えてからの検知であると遅い場合などがある。ゆえに今回は閾値による変化点検知ではない、自己回帰型モデルを用いた変化点検知について説明する。

■自己回帰モデルを用いた変化点検知　時系列データに対し、その特徴を表現する時系列モデルの一つに、自己回帰モデルがある。自己回帰モデル（AR モデル）については、§3.3.2 で詳しく述べるが、自己回帰モデルを用いた変化点検知とは、時刻 t 以前の時系列データを部分的に用いて、自己回帰モデルという時系列モデルにあてはる。そしてモデルから算出された時刻 t における期待値と、時刻 t における実際の観測値との差異から、時刻 t が変化点であるのかを判定する方法である。

以下では自己回帰モデルを用いた変化点検知のアルゴリズムをの一例である `changefinder` を（`https://pypi.org/project/changefinder/`）、数式を用いて具体的に説明する。まず、時刻 t までの時系列データ $Y_{1:t-1} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{t-1}\}$ に対し、AR モデルを構築する。この時構築された AR モデルから得られる確率密度関数を $p(\boldsymbol{y}_t|Y_{1:t-1})$ とする。この確率密度関数に対し、時刻 t のデータ $\boldsymbol{y}_t$ における変化点スコア $\mathrm{score(y_t)}$ を

$$\mathrm{score(y_t)} = -\log p(\boldsymbol{y}_t|Y_{1:t-1}) \tag{3.1.2.11}$$

とする。

次に式（3.1.2.11）より求まる変化点スコアを、平滑化する。具体的には、任意の正の整数 L のウィンドウ幅内の時系列データに対して、変化点スコアを求め、その平均を z_t を

$$z_t = \frac{1}{L}\sum_{i=t-L+1}^{t} score(\boldsymbol{y}_t) \tag{3.1.2.12}$$

のように求める。これより、変化点スコアの移動平均系列 $Z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ を構成する。この作業を平滑化と呼び、変化点検知を検出した時の、その後の検出への影響を小さくするために行う。加えてこのときのウィンドウ幅が大きくなると、より幅のある変化点検知がが可能となる。

その後 z_t に対して AR モデルを再構成する。そしてこの操作で得られる確率密度関数 $q(\boldsymbol{z}_t|Z_{1:t-1})$ を用いて、時刻 t における変化点スコア $\mathrm{Score}(t)$ を

$$\mathrm{Score}(t) = -\log q(\boldsymbol{z}_t|Z_{1:t-1}) \tag{3.1.2.13}$$

と計算する。最後に時刻 t における変化点スコアに対し、閾値 s を設定し、

$$\mathrm{Score}(t) > s \tag{3.1.2.14}$$

となる時刻を、変化点として検出する。

今回は変化点検知に際し、`changefinder` という Python のライブラリを用いた。これは自作した関数で行うと、計算量が多く、計算に用いる計算量が多くなってしまったので、既存のものを用いた。`changefinder` では、上記と同様な手順を踏み、変化点検知を行う。ただし変化点検知の高速化のため、AR モデルによるモデルのあてはめを高速化した、Sequentially Discounting AR model(SDAR) モデルを AR モデルの代わりに利用している。`changefinder` 三つのパラメタがあり、それぞれのパラメタは AR モデルの次元、L の大きさ、そして SDAR モデルのパラメタを定められる。

3.1.3　異常検知の評価

異常検知器の精度を評価をする際、正常・異常のクラスタラベルごとの正解率を、そのまま評価として用いてはいけない。(6)(17) まず正常・異常の判断をする際、異常であるのに正常と判断することは、正常であるのに異常と判断するよりも大きな問題となる。ゆえに、二つのカテゴリの正解率を、同等な重要度で評価することは、間違ったモデル選択につながる可能性がある。次に、異常検知を行う際、対象のデータセットにおいて、異常のサンプルと正常のサンプルの比率は大きく偏っている。例えば、99 %が正常なデータであるデータセットの場合、すべてのサンプルに対して正常であると判断することで、誤り率は 1 %に抑えられるが、これは明らかに有用な判別器でない。これより、異常検知を行う判別器の評価は、正解率以外の指標が必要となる。そこで、混合正規行列 (confusion matrix) や ROC 曲線、AUC 値、F 値などを用いて異常検知器の性能を評価する。

■混合行列　異常検知において、全体の正解率のみに着目することでは、異常検知器の性能を正しく判断することはできない。そこで、混合行列 (ConfusionMatrix) を使うことで、正解率以外の指標を用いることができ、検知器の全体的な評価をしやすするなる。まず異常カテゴリを真、正常カテゴリを偽としたとき、判別器の結果は以下の 4 パターンが考えられる。

I. True Positive(TP)　真を真と判断したケース
II. False Positive(FP)　偽を真と判断したケース
III. True Negative(TN)　偽を偽と判断したケース
IV. False Neagtive(FN)　真を偽と判断したケース

これらのケースに該当した結果を表にしたものを、混合行列という。混合行列は表 (3.1) のように表される。このとき、混合行列の項目を用い、検知器の性能は以下の四つの尺度で評価することができる。

I. 正解率：全サンプルの中でラベルを正しく判別できた割合

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{3.1.3.1}$$

II. 適合率：真と分類されたサンプルの内、実際に真であるサンプルの割合

$$\frac{TP}{TP+FP} \tag{3.1.3.2}$$

III. 再現率・陽性率：本来は真と判別すべきサンプルの内、実際に真と判別されたサンプルの割合

$$\frac{TP}{TP+FN} \tag{3.1.3.3}$$

IV. 特異度：本来は偽と判別すべきサンプルの内、実際に偽と判別されたサンプルの割合

$$\frac{TN}{TN+FP} \tag{3.1.3.4}$$

この四つの指標の中で、異常検知器の際、重要なのが、再現率である。再現率は、異常と検知すべきサンプルのうち、実際に異常と判断されたサンプルの割合を示す。これが低いと、異常であるのに異常と判断されない異常検知器となる。ただし適合率と再現率はトレードオフな関係であり、どちらかが上がると、どちらかが下がる傾向にある。適合率は、異常と判断されたサンプルのうち、実際に異常である割合である。適合率が低いと、異常でないのに異常と判断する例が多くなってしまう。これもまた、良い異常検知器とは言えなくなる。ゆえに、両者の値を考慮した評価指数が必要となる。その中で代表的なものが F 値と AUC 値である。

表 3.1. 混合行列

		予想結果	
		真	偽
実際のクラス	真	TP	FN
	偽	FP	TN

■F 値　F 値は、適合率と再現率の調和平均で計算され、

$$F\ 値 \quad = \quad \frac{2}{\frac{1}{再現率}+\frac{1}{適合率}} = \frac{2\times 再現率 \times 適合率}{再現率 + 適合率} \tag{3.1.3.5}$$

と表す評価指数である。F 値がの最大値は 1.0 であり、これは適合率と再現率がともに 1.0 の場合である。

■ROC 曲線・AUC 値　ROC 曲線は、各サンプルに対し異常値が算出された状態で、異常と判断する閾値を $-\infty \sim \infty$ まで変化させたときの、陽性率と偽陽性率を二次元にプロットした図である。ただし偽陽性率とは $1-$ 特異度 であらわされる尺度であり、正常サンプルのうち、異常と判断された割合である。これにより異常度に関する閾値によらない、全体的な判別機の性能を評価することができる。図（3.5）が ROC 曲線の例と、ランダムの場合の ROC 曲線の例である。精度がよい検出器は、左上の偽陽性率が 0 かつ、陽性率が 1 である点を通るような曲線となる。

ROC 曲線は直感的に判別性能を評価できるのに対し、これを定量的に評価する場合は AUC 値 (area under the ROC curve) を用いる。AUC 値は ROC 曲線の下側の面積で定義される。AUC 値はランダムに真、偽を出力する場合、AUC 値は 0.5 となり、1 に近いほど判別精度がよいと判断できる。

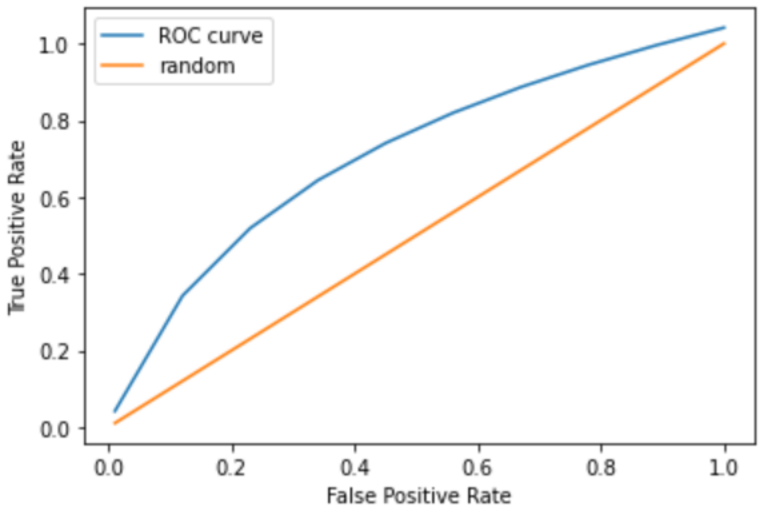

図 3.5. ROC curve の例

3.1.4　機械学習的手法と統計的手法

今回 §3.2 の異常検知アルゴリズムを踏まえ、異常検知器を開発にするにあたり、機械学習的手法と統計的手法を活用した。§3.2、§3.3 では今回用いたそれぞれの手法について説明する。

基本的に統計的手法とは、与えられたデータに対し、モデルの構築や、仮説検証を行うことで、データの説明を行う手法である。ゆえに統計的手法は、ある程度直感的に理解できる説明変数で構成されたデータを分析するのに適している。例えば既存の測定データをある確率モデルにあてはめ、未知の値を予想したり、値の実現確率を算出するなどの手法を行う。

一方機械学習的手法は、データの説明ではなく、データの背景のルールを学び、予想することを主軸とした手法である。ゆえに機械学習的手法は、直感的には理解しづらいルールに従うデータに対し分析をするのに用いられることが多い。例えば既存の測定データをわける判別平面をいくつものパラメータを変えながら調べたり、クラスタリングをしたりすることができる。

今回は主に非時系列データであるノイズスペクトルデータに対しては、機械学習的手法を用いた外れ値検知、時系列データに対しては統計的手法を用いた変化点検知を行った。これはノイズスペクトルデータは、非時系列データはひとつのモデルにあてはめることが難しかったので、機械学習的手法を用いた外れ値検知を用いた。また温度時系列データは、変化点検知でのデータは当てはめる際に、時系列モデルが予測しやすかったから統計的手法を用いた。

3.2 機械学習的手法

3.2.1 機械学習について

機械学習とは、与えられたデータの背景にあるルールを学習し、学んだルールに基づき、未知のデータの値を予想したり、データが属するカテゴリの判別を行う手法である。本論文で用いた機械学習のアルゴリズムは大きく、教師あり学習と、教師なし学習に分けられる (18)。

教師あり学習では、ユーザーは入力データと望ましい出力のペア群を、コンピュータに与える。その与えられたペア群から、コンピュータは未知の入力データに対する、望ましい出力を生成するルールを学習する。教師あり学習の主な用途は、回帰と分類である。回帰とは、コンピュータが、学習したデータをもとに、入力されたデータから、連続する出力を予測する分析である。分類とは、コンピュータが、学習したデータをもとに、入力されたデータを、複数のカテゴリに分ける分析である。

一方、教師なし学習では、コンピュータには入力データのみが与えられ、望ましい出力データが与えられない。その入力データから、コンピュータは学習を行い、データ間の距離や分散などをもとに、目的に応じた理想的な出力を生成するルールを発見する。教師なし学習は、データに隠されたルールを見つけ、膨大なデータを、人間やほかの機械学習アルゴリズムにとって、より分かりやすい新しいデータ表現にすることができる。そのため教師なし学習は、教師あり学習の前処理として利用することも多い。教師なし学習の主な用途は、次元削除とクラスタリングである。次元削除とは、多次元のデータをより少ない次元のデータに、なるべく元の情報を損なわずに変換することである。クラスタリングとは、データセットのサンプル間距離などの情報から、入力サンプルを複数のカテゴリに分類することである。

3.2.2 代表的なアルゴリズム

機械学習的手法における異常検知器の作成にあたり、Python における `scikit-learnhttps://scikit-learn.org/stable/index.html` ライブラリを利用した。どのアルゴリズムを用いるかは、目的やデータセットに基づき scikit-learn が作成した図 (3.6) の Estimator Map から判断した。以下では、本研究で用いた機械学習のアルゴリズムについて説明する。ただし本研究では、分類とクラスタリング、次元削除を用い、回帰アルゴリズムは用いなかった (18)(17)(15)

。

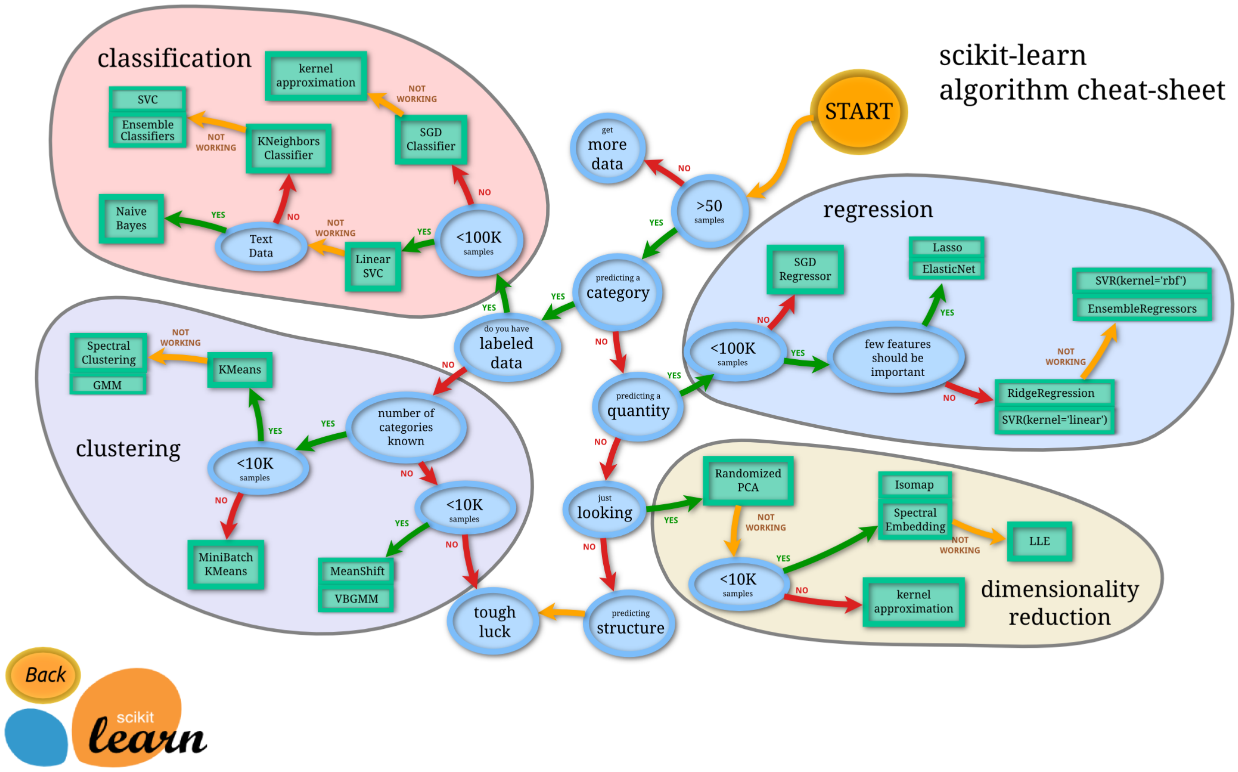

図 3.6. EstimatorMap`scikit-learn`(`https://scikit-learn.org/stable/index.html`)

分類

分類（Classification）では、訓練データと訓練データの理想的なカテゴリラベルが与えられ、それぞれのカテゴリへ分類するルールを学習する。その後、分類したいデータを、学習したルールに従い、それぞれのカテゴリに分ける。分類では、主に識別平面を区切りにし、識別平面よりも上にあるデータと下にあるデータで分けることで、カテゴリの分類を行う。この識別平面は、線形平面と非線形平面の場合があり、線形判別器で分類がうまくいかない場合、非線形分類器を用いることが多い。今回は線形判別器を用いたため、以下では線形判別器と基本的な線形判別平面の定め方について説明する。

線形分類器を用いるとき、識別平面は入力データが n 次元空間の場合

$$f(\boldsymbol{X}) = u_1x_1 + u_2x_2 + ... + u_nx_n + b = 0 \tag{3.2.2.1}$$

となる点 $\boldsymbol{X}$ の集まりとして表される。ただし、$f(\boldsymbol{X})$ は実数関数である。このとき、識別平面をもとめることは、$f(\boldsymbol{X})$ の係数 $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ と切片 b を求めるということである。そして識別平面を求めるために、識別平面を求めるための最適化問題を解くことでパラメータを求める。ただ、このパラメータを定める際、訓練データに完全に適したパラメータを用いると、過学習を引き起こす可能性がある。過学習とは、訓練データに適合しすぎて、未知のテストデータに対しての精度が悪くなる現象である。ゆえに識別平面のパラメータを決める際は、未知のテストデータに対する識別能力をあげるための、最適化問題の式変形を行う。これを正則化という。基本的に、線形分類器のアルゴリズムの違いは、どのような最適化問題を解くか、どのように最適化問題を解くかによって異なる。

■Linear SVC　LinearSVC とは、サポートベクターマシン (SVM) のアルゴリズムを用いて、線形判別平面定め、分類を行うアルゴリズムである。LinearSVC のメリットは、多次元データでの解析が高速に可能であることであ

る。デメリットは、特徴量がサンプルよりも多すぎると過学習を起こす可能性があり、正規化の強さに注意が必要なことである。

まず SVM とは、判別平面と最も近いデータ点の距離が最大になるような最適化問題を解き、その問題の双対問題を解くことで判別平面を定め、分類をする方法である。以下では SVM の解くべき最適化問題の導出、最適化問題の解き方を説明する。

まず初めに最適化問題の導出をする。クラス G_1、G_2 のいずれかに属するサンプル $\bar{x} \in \chi \subset \mathbb{R}^p$ があり、これがクラス G_1 に属する場合には、クラスラベル $y = 1$、クラス G_2 に属する場合は、$y = -1$ という値をとるとする。こうしたサンプル $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ をもちいて、新たに観測したサンプルが属するクラスを予想する予測器を構成する問題を考える。このとき、識別平面を定めるにあたり、元のデータ空間 χ における超平面ではうまく 2 クラスの分離ができない場合でも、適切な次元空間 H を考えて、元のサンプル $\boldsymbol{x}$ を変換した $\phi(\boldsymbol{x}) \in H$ なら分離がうまくいくことがあることがある。これを利用し、空間 H において

$$f(\boldsymbol{x}) = \boldsymbol{\omega}^T\phi(\boldsymbol{x}) + b \tag{3.2.2.2}$$

なる判別関数を考える。そして、この実数関数 $f(\boldsymbol{x})$ を用い、$f(\boldsymbol{x}) > 0$ なら $y = 1$、$f(\boldsymbol{x}) < 0$ なら $y = -1$ として判別をする。しかし、このときカテゴリを分ける超平面は一意に定まらない。なぜなら図（3.7）のように、複数の超平面が想定されるからである。そこで超平面を一意に定めるため、各クラスの学習データから超平面までの最短距離を最大化する平面を分離超平面とし、それを判別平面とする。よってこの問題は、学習データから、超平面の距離を d とすると、

$$\begin{gathered}\max_{\boldsymbol{\omega},b}\min_{i=1,2,\ldots n} d_i = \max_{\boldsymbol{\omega},b}\min_{i=1,2,\ldots n}\frac{|\boldsymbol{\omega^T\phi(x_i)+b}|}{||\boldsymbol{\omega}||}\\ \text{s.t.}y_i(\boldsymbol{\omega}^T\phi(\boldsymbol{x}_i)+b) > 0, i = 1, 2, ..., n\end{gathered} \tag{3.2.2.3}$$

と表される。ここで、超平面のパラメータ $\boldsymbol{\omega}, b$ を同じだけ定数倍しても、最適化関数の分母と分子で打ち消しあい、s.t. 以下の制約条件の符号も変わらないため、この問題に影響はない。そこで、$\min_i|\boldsymbol{\omega}^T\phi(\boldsymbol{x}_i)+b)| = 1$ という条件を加えると、式（3.2.2.3）は

$$\begin{gathered}min_{\omega b}||\omega||\\ \text{s.t.}y_i(\omega^T\phi(\bar{X}_i)+b) = 1\end{gathered} \tag{3.2.2.4}$$

と変形できる。これを解くことで、判別平面は求まるが、$y_i(\omega^T\phi(\bar{X}_i)+b) = 1$ が成立するためには、線形判別平面によって完全にデータが分離される必要がある。しかし実際のデータにおいては、完全な線形分離が可能とは限らない。ゆえに、なるべくこの関係が満たされるように、最適化問題を解くという条件に書き換える必要がある。そこで $y_i(\omega^T\phi(\bar{X}_i)+b) = 1$ に対する、損失関数 L を定義し、損失関数がなるべく小さくなるようにパラメータを決める必要がある。これをまとめると、SVM では、以下の式を満たす最適化問題を解くことで、カテゴリに分類することができる。

$$min||\omega|| + C\sum_{i=1}^{n}L_i \tag{3.2.2.5}$$

そしてこれを一般化すると、以下のような式を解く。

$$min_{\omega,b}\Omega(\omega) + C\sum_{i=1}^{n}L(f(X_i), y_i) \tag{3.2.2.6}$$

ただし C を正則化パラメータ、f を判別関数とする。このときの Ω は本来 $|\omega|$ を最小にするための関数である。このとき、$|\omega|$ をどの程度の強さで最小にするかは、Ω の関数によって定められる。具体的には Ω は

$$\Omega(\omega) = \frac{1}{p}||\omega||^p \tag{3.2.2.7}$$

とあらわされ、scikit-learn 上では、loss パラメータを l1 にするとき $p=1$、l2 にするとき $p=2$ となる。

また損失関数についても、どの程度の強さで判別平面を教師データに適応させたいかで、選ぶ式を選ぶことができる。例えば損失関数 L はヒンジ関数や修ヒンジ関数などがある。損失関数がヒンジ関数の場合は、

$$L(m) = \max(1-m, 0) \tag{3.2.2.8}$$

という式になる。また損失関数が修正ヒンジ関数の場合は

$$L(m) = \max(1-m, 0)^2 \tag{3.2.2.9}$$

となる。これらの式の違いについては、§A.2 を参照してほしい。scikit-learn では、`penalty` パラメータを `penalty` が `hinge` にするとき、ヒンジ関数を、`penalty` を `squared_hinge` にする場合は修正ヒンジ関数となる。このような最適化問題を解くことで、ω、b を定める。以上が SVM の満たすべき、最適化問題の導出である。

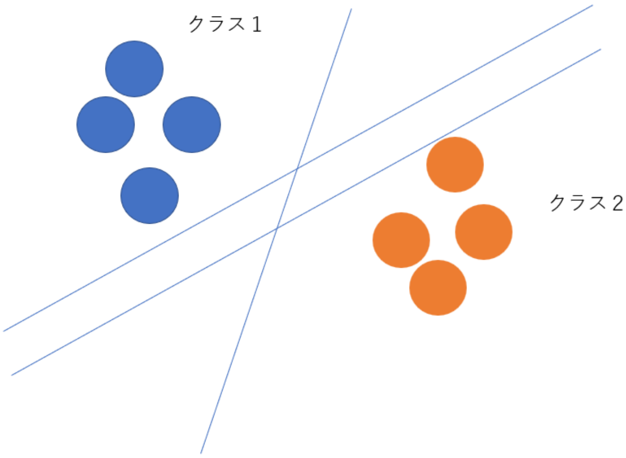

図 3.7. SVM の概要図

次に最適化問題の解き方を説明する。基本的には、この問題に対して、少しづつパラメータを変えて最適なものを選ぶ。しかし LinearSVC では、この最適化問題を双対問題を解くこともできる。双対問題とは、本来解きたい計算コストが大幅にかかる問題の代わりに解く、計算コストが良い問題である。今回の解きたい問題は式（3.2.2.5）より

$$S(\omega, b) = \frac{1}{2}||\omega||^2 + \sum_{i=1}^{N} \alpha_i \{y_i(\boldsymbol{\omega}^T \phi(\boldsymbol{x}_i) + b) - 1\} \tag{3.2.2.10}$$

としたときの L を最小とするパラメータの導出である。このとき $L(\omega, b)$ を最小にするパラメータ ω、b は偏微分を行うと、

$$\omega = \sum_{i=1}^{N} \alpha_i y_i \phi(\boldsymbol{x}_i) \tag{3.2.2.11}$$

、

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.2.2.12}$$

となり、これを解けばよい。これらをラグランジュ関数に代入すると以下のような式となる。つまりこの時二つのパラメータを定めるためには、これを満たす α を算出するべきであり、この双対問題は

$$S(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) \tag{3.2.2.13}$$

を最大にする α を求める。以上が LinearSVC の最適化問題の解き方である。さらにこの問題を解く際にも、カーネル法を用いることで、計算量を少なくすることが出来る。具体的には、

$$K(x_i, x_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) \tag{3.2.2.14}$$

は高次元になるほど、計算量が増える。そこでこれを直接計算する代わりに、カーネル関数で近似をする。カーネル関数は複数種類あるが、LinearSVC では線形カーネル関数を用いて

$$K(x_i, x_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{3.2.2.15}$$

として解く。以上が LinearSVC の最適化問題の解き方である。

このとき LinearSVC ではまた表 (3.2) は LinearSVC の scikit-learn における主要なパラメータ一覧である。

表 3.2. LinearSVC の主要なパラメータ

パラメータ	種類	内容
penalty	{l1,l2}	正規化の種類
loss	{'hinge', 'squared_hinge'}	損失関数の種類
tol	float	止める基準
C	float	正規化パラメータ、正規化の強さの基準
max_iter	int	反復回数の最大値
dual	bool	双対問題を解くか否か

■Stochastic Gradient Descent　Stochastic Gradient Descent(SGD) は、確率的勾配降下法をもちいて、判別平面を定め、分類を行うアルゴリズムである。SGD のメリットは、大きなスケールのデータ処理に適し、特に疎で巨大なパラメータやサンプルを持つなデータ処理が得意である。デメリットは、データのスケーリングに繊細であることや、パラメータが多くパラメータチューニングに時間がかかることである。

まず SGD とは、確率的勾配降下法を用いて最適化問題を解き判別平面を定め、分類をする方法である。以下では SGD の解くべき最適化問題の導出、最適化問題の解き方を説明する。

まず初めに最適化問題の導出をする。SGD は

$$E(\omega, b) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(y_i)) + \alpha R(\omega) \tag{3.2.2.16}$$

が最小となるように線形判別平面を定め、分類を行うアルゴリズムである。このとき、L は損失関数、R は正則項である。このとき L は教師データに対する損失関数であり、教師データに対し分類方法を間違えるとスコアが増えていく。そして $R(\omega)$ は正則項である。このときの正則化関数については、§A.3 を参照してほしい。このときの正則項と損失関数は目的に応じて変更できる。

まず SGDClassifier における、損失関数 L のなかで主要なものを以下に記す。

I. Hinge

$$L(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) \tag{3.2.2.17}$$

II. Log Loss

$$L(y_i, f(x_i)) = \log(1 + \exp(y_i f(x_i))) \tag{3.2.2.18}$$

III. Squared Error

$$L(y_i, f(x_i)) = \frac{1}{2}(y_i - f(x_i))^2 \tag{3.2.2.19}$$

同様に正則化を担うペナルティ関数も以下に記す。SGDClassifier でのペナルティ関数は三種類ある。一つ目が L1 正則化、二つ目が L2 正則化である。これは以下のような式で表され

$$R(\omega) = \frac{1}{p}||\omega||^p \tag{3.2.2.20}$$

p が 1 のときは L1 正則化、p が 2 のときは L2 正則化となる。三つ目が、Elastic Net という正則化もあり、

$$R(\omega) := \frac{\rho}{2}\sum_{j=1}^{n}\omega_j^2 + (1-\rho)\sum_{j=1}^{m}|\omega_j| \tag{3.2.2.21}$$

と表せられる。

次に最適化問題の解き方を説明する。SGD Classifier は最適化問題を最小とするパラメータ探索のため、確率的勾配降下法を用いる。確率的勾配降下法とは、与えられたデータに対してコストが最小になるように、モデルパラメータを少しづつ操作し、モデルをデータに対して最も適したパラメータに収束させる方法のことを指す。具体的な方法は、まず初期値のパラメータにおいての、損失関数の勾配を求める。そして勾配の符号方向に、ランダムに次の点を取り、勾配がゼロになるパラメータを求める。そして勾配が 0 になる点でのパラメータを採用する。確率的勾配降下法の際のパラメータ x は

$$x := x - \zeta\frac{df(x)}{dx} \tag{3.2.2.22}$$

のように更新され、ζ はランダムに定まる。図（3.8）は確率的勾配降下法による、候補パラメータ x の更新の概要を図示したものである。

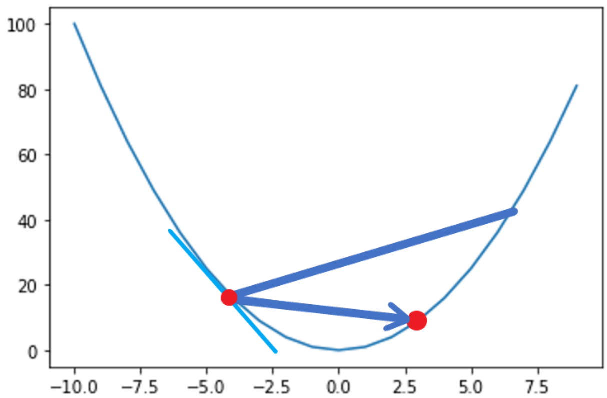

図 3.8. 確率的勾配降下法の概要

また表 (3.3) は SGDClassifier の scikit-learn における主要なパラメータ一覧である。

■KNN 法　k-Nearest Neighbors 法（KNN 法）とは、LinearSVC、SGD とは異なり、判別平面を定めず分類を行う方法である。KNN 法では、テストデータサンプルに対し、近接する点 k 個から、そのテストデータサンプルが属するカテゴリを分類する。KNN 法のメリットは、アルゴリズムが簡単で理解しやすいことである。デメリットはアルゴリズムに近傍のサンプルとの距離を用いるので、次元数が多いデータの場合、距離がどれも似たようなものとなり、正確な分類ができないことである。

KNN 法のアルゴリズムでは、初めに任意のテストデータを一つ選び、そのテストデータに近い k 個の訓練データを探す。その k 個の訓練データの中で、一番多いカテゴリをそのデータのカテゴリとする。図（3.9）が KNN 法のアルゴリズムを図示したものである。

表 3.3. SGDClassifier の主要なパラメータ

パラメータ	種類	内容
penalty	{l1,l2}	正規化の種類
loss	{'hinge', 'log_loss', 'log', 'modified_huber','squared_hinge', 'perceptron', 'squared_error', 'huber', 'epsilon_insensitive', 'squared_epsilon_insensitive'}	損失関数の種類
alpha	float	正規化の強さ
tol	float	止める基準
max_iter	int	反復回数の最大値

scikit-learn では、k の値はパラメータによって決めることが出来る。しかし基本的にカテゴリが 2 つの場合は、k の値を奇数にする。これはあるテストデータに対し、近接するカテゴリのデータ数が同数となるのを防ぐためである。表（3.4）は KNN 法の scikit-learn における、主要なパラメータ一覧である。

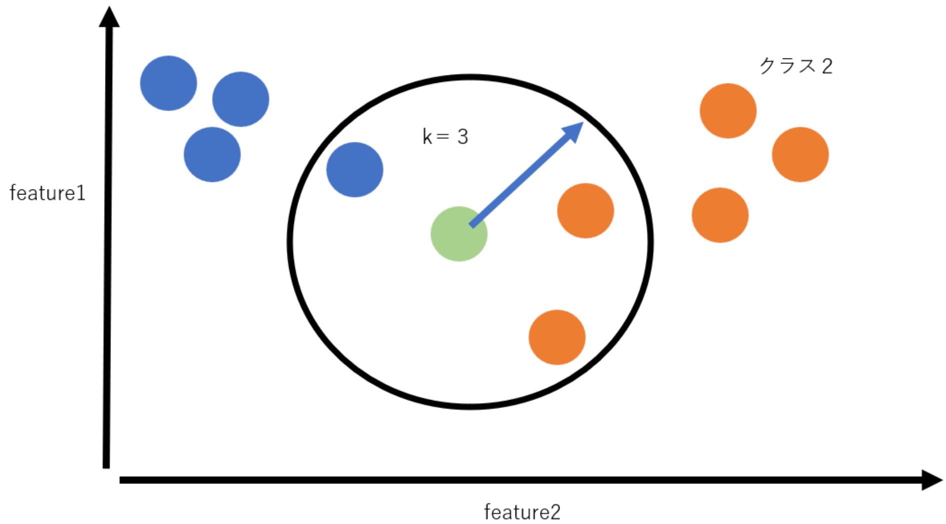

図 3.9. knn 法の概要図

表 3.4. KNN の主要なパラメータ

パラメータ	種類	内容
n_neighbors	int	近接するサンプルの数
weights	{'uniform', 'distance'} or callable	データ点の重み

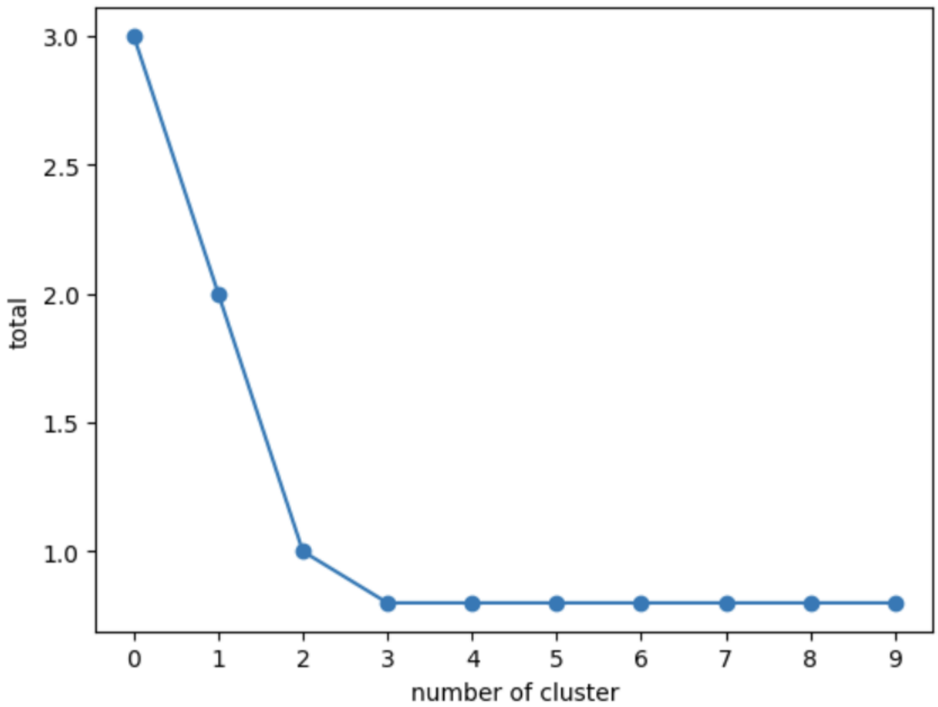

図 3.10. エルボー法の例

クラスタリング

クラスタリング（Clustering）は、クラスラベルの情報が与えられない状況で各データに対する特徴量の類似度や距離に基づき、データをいくつかのグループに分類する方法である。クラスタリングの結果は、用いるアルゴリズムや採用するデータ同士の距離に大きく影響を受けるため、データの前処理やデータの性質に応じた距離を採用することなどが重要である。またクラスタリングの際には、クラスタの数をユーザが決める必要がある。これに関して明確な方法は確立されていないが、代表的な方法としてエルボー法がある。

エルボー法とは、クラスタリングをする際、最適なクラスタ数を求める手法である。エルボー法では、各クラスタの重心点とクラスタ所属の各点の距離の和をプロットする。この例が図（3.10）である。このとき、クラスタの数が増えれば、それぞれの点に近いクラスタが存在するようになるので、距離の総和は減少する。そしてある一定以上クラスタを増やしたとき、この減少スピードが緩やかになる。この変わり目のクラスタ数の時、最適なクラスタ数と解釈する。例えば図（3.10）の場合は、2 個のクラスタが最適と判断する。

■K-means 法　K-means 法は、各クラスタの重心の距離が最大になるように、クラスタリングを行う方法である。K-means 法のデメリットは、クラスタに等方性があることを仮定しているので、細長いクラスタや不規則な形状のクラスタを想定する場合は、精度の高いクラスタリングができない。以下では K-means 法のアルゴリズムについて説明する。

今回は K 個のあらかじめ指定したクラスタ数に、各サンプルを分類する場合を考える。このとき各クラスタのサンプルと、そのクラスタの重心からの距離を、クラスタごとに足しあげたものを J とするとき、J を最小化するクラスタ割り当てを採用する。この問題は、組み合わせ最適化問題であり、厳密な解を求めるのは難しい。そこで、K-means 法では、以下のステップを行い、最適解を探す。

1. 入力サンプルをプロットし、そのうちランダムに K 個の点を抽出する。

2. 抽出した点にクラスタ 1, クラスタ 2... などラベリングをし、クラスタ k での点を S_k とする。そして、それぞれの入力サンプルを、S_k（$k = 1, 2, , ...K$）の中で、S_k との距離が一番近い点のクラスタに割り当てる。このとき、クラスタ k に属するサンプルの集合 C_k は以下のようにあらわされる。

$$C_k = \{\boldsymbol{x}_i | arg \min_{l \in 1, \ldots K} d(\bar{\boldsymbol{x}}_l, \boldsymbol{x}_i) = k\} \tag{3.2.2.23}$$

3. 現在のクラスタの割り当てから、各クラスタの中心を計算する。

$$\bar{\boldsymbol{x}_k} = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i \tag{3.2.2.24}$$

4. 2 で定めた S_k を更新し、$S_k = \bar{\boldsymbol{x}_k}$ とする。
5. 2-4 の過程を繰り返し、重心の移動距離がなくなるとき、終了とする。

scikit-learn では過程 5 において、実際に行う際は、重心が移動しなくなるまで行うのは難しいので、繰り返し上限回数に達するか、重心の移動距離が十分に小さくなったら終了する。加えて K-means 法の結果は、初期クラスタ中心の選び方に依存する。ゆえに scikit-learn を用いるときは、再現性を確保するため、初期クラスタ中心の選び方を、固定することが必要である。図（3.11）は k-means 法のアルゴリズムを図示したものである。

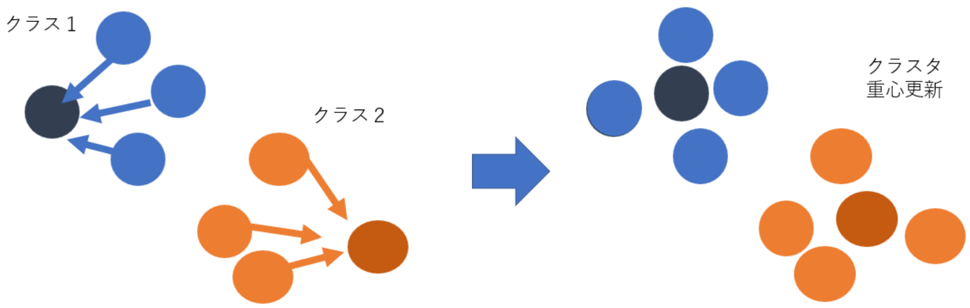

図 3.11. kmeans 法の概要図

■MiniBatchK-means 法　MiniBatchK-means 法とは、K-means 法を簡略化し、計算時間を短縮した方法である。具体的には、K-means 法ではすべてのデータで、重心を更新したが、MiniBatchK-means 法では batch_size 個のまとまりごとに、重心を更新する。batch_size が大きいと、計算時間は短縮されるが精度は悪くなる。表（3.5）は MiniBatchK-means の scikit-learn における、主要なパラメータ一覧である。

表 3.5. MinibatchK-means の主要なパラメータ

パラメータ	種類	内容
n_cluster	int	クラスタの数
batch_size	int	重心を更新する際にまとめて行うデータ数

次元削除

次元削除とは、元データが存在する高次元空間から、元データの情報を損なわず、低次元空間に変換する方法である。次元削除の主な目的は、情報の縮約による過学習の防止や計算コストの削減、可視化、以降の処理に適したデータ表現の発見である。次元削除には、T-SNE や Singular value decomposition(SVD) などいくつかの方法があるが、今回は Principal componet analysis(PCA) について説明する。

■PCA　Principal componet analysis(PCA) とは、主成分分析のことを指し、多変量のデータを少数個の主成分と呼ばれる合成変数で記述する方法である。具体的には、データセットの特徴量を、相互に統計的に関連しないように回転させ、その回転後の特徴量の中から、データを説明するのに重要な一部の特徴量を抜き出す。主成分分析は統計的手法にふくまれることが多いが、今回は scikit-learn を用いて実装したので、機械学習のカテゴリに書く。以下では主成分分析の数学的背景とアルゴリズムを説明する。

まず、p 個のデータが含まれるデータセット $X = \{\boldsymbol{x}_1, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_p\}$ があり、それぞれのデータは n 次元のデータであるする。このとき主成分分析前のある n 次元の変数を $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ とし、大きさ 1 のベクトル $\boldsymbol{y}_i (i = 1, 2, \ldots, n)$ を用いて、

$$\begin{aligned}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} &= m_1 \begin{pmatrix} y_{1,1} \\ y_{2,1} \\ \vdots \\ y_{n,1} \end{pmatrix} + m_2 \begin{pmatrix} y_{1,2} \\ y_{2,2} \\ \vdots \\ y_{n,2} \end{pmatrix} + \cdots + m_n \begin{pmatrix} y_{1,n} \\ y_{2,n} \\ \vdots \\ y_{n,n} \end{pmatrix} \\ &= m_1 \boldsymbol{y}_1 + m_2 \boldsymbol{y}_2 + \cdots + m_n \boldsymbol{y}_n \end{aligned} \tag{3.2.2.25}$$

と表せられるとする。この時のデータセット X が $\boldsymbol{y}_i$ に対しての分散が大きいとき、データセット X に対して、$\boldsymbol{y}_i$ 成分の影響が大きいといえる。具体的には、データセット X に含まれる任意の変数 $\boldsymbol{x_j}$ に対し、$\boldsymbol{x_j}$ に対する y_i の係数を m_{ji} とする。このとき集合 $M = \{m_{1i}, m_{2i}, \ldots, m_{pi}\}$ の分散が大きいとき、データセット D に対して、$\boldsymbol{y}_i$ 成分の影響が大きいといえる。このように分散が最大となる $\boldsymbol{y}_i$ 成分を求めるのが主成分分析である。ここで $\boldsymbol{x}_j = \{x_{1j}, x_{2j}, \ldots, x_{nj}\}$ の合成変数 m_i は任意の $\boldsymbol{u} = (u_{1,i}, u_{2,i}, \ldots, u_{n,i})$ を用いて

$$m_i = u_{1,i}x_{1j} + u_{2,i}x_{2,j} + \cdots + u_{p,i}x_{p,j} \tag{3.2.2.26}$$

と表されるとする。このとき m_i の分散 $V[m_i]$ は

$$\begin{aligned} V[m_i] &= \boldsymbol{u}^T S \boldsymbol{u} \\ &= \boldsymbol{u}^T \begin{pmatrix} s_{1,1} & \cdots & s_{1,p} \\ \cdots & \cdots & \cdots \\ s_{p,1} & \cdots & s_{p,p} \end{pmatrix} \boldsymbol{u} \end{aligned} \tag{3.2.2.27}$$

と表されるので、主成分分析は $V[y_i]$ の最大化問題として定式化される。そして、$V[y_i] = \lambda_i$ とすると

$$S\boldsymbol{u} = \lambda_i \boldsymbol{u} \tag{3.2.2.28}$$

と表されるので、固有値問題に帰着できる。ただし S は標本分散共分散行列（A.4）である。また、λ は、$\boldsymbol{y}_i$ 成分に対する、分散を表しているので、

$$c_j = \frac{\lambda_j}{\sum_{i=1}^{n} \lambda_i} \tag{3.2.2.29}$$

とすると、c_j は $\boldsymbol{y}_j$ 成分の、全体に対する寄与率を表す。寄与率が高いほど、元のデータに影響力を与えているので、PCA により次元削除をする場合は寄与率が高い方から、$\boldsymbol{y}_j$ 成分を採用する。

3.2.3　機械学習の実用

機械学習で、予測の精度を向上させるためには、データのスケール変換、凡化性能の向上を行うことが必要不可欠である。加えて、教師あり学習を行う際は、教師データの数が各カテゴリに対し、不均一な場合、不均一なデータを均一する作業が必要となる。以下では、データのスケール変換や凡化性能の向上、不均一データの均一化で行う作業について、説明する。

データのスケール変換

機械学習のアルゴリズムのなかには、データのスケール変換に敏感なものがある。それらのアルゴリズムでは、各特徴量に対し同一化スケールとなるように変換する必要がある。この際、それらのアルゴリズムに適したデータ表現にするためには、標準化や正規化を行うとよい。以下では標準化と正規化について説明する。ただし説明では、あるデータセット $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ に対し、i 番目の特長量 $X_i = \{\boldsymbol{x}_{1,i}, \boldsymbol{x}_{2,i}, \ldots, \boldsymbol{x}_{n,i}\}$ のスケール変換を行うとする。

標準化とは個々の特徴量の平均が 0、分散が 1 になるように変換し、すべての特徴量の大きさをそろえる変換方法である。このとき、任意のデータの $x_{j,i}$ を標準化により $x'_{j,i}$ に変換する際は、

$$x'_{j,i} = \frac{x_{j,i} - \bar{X}_i}{\sigma} \tag{3.2.3.1}$$

のように行う。ただし $\bar{X}$ はデータセット X_i の標本平均、σ はデータセット X_i の標準偏差である。

一方正規化とは、個々の特徴量を 0 から 1 の間に収めるようにする、変換方法である。このとき、任意のデータの $x_{j,i}$ を正規化により $x'_{j,i}$ に変換するには、

$$x'_{j,i} = \frac{x_{j,i} - \min(X_i)}{\max(X_i) - \min(X_i)} \tag{3.2.3.2}$$

のように行う。ただし $\max(X_i)$ とは X_i データセットの中の最大値であり、$\min(X_i)$ とは X_i データセットの中の最小値である。

凡化性能の向上

機械学習において訓練データからモデルを作成するとき、気を付ける必要があるのが過学習である。そして過学習を起こさず、未知のデータに対しても、適応するモデルを、凡化性能が高いモデルという。この凡化性能を評価する、モデルの評価方法の一つが交差検証法である。凡化性能が高いモデルを選ぶためのパラメータやモデル選択方法で用いられるのが、グリッドサーチである。

■交差検証法　機械学習を用いて得られたモデルが、どの程度精度を持つのかを調べる方法の一つである。モデルの精度を調べる際、モデルを作成するときに学習させたデータを用いて、精度を調べても意味がない。なぜなら、モデルは学習データの精度を最大化するように作られており、学習させたデータに対して精度が高いことは、当たり前であるからだ。このとき、未知のデータにもモデルが有効であることを示すには、学習したデータとは異なるデータセットを用いて、検証する必要がある。このとき、よく用いられる方法が交差検証法である。交差検証では、各群のデータを K 分割し、そのうちの 1 セットをテスト用とし、それ以外のデータを用いて学習をする。1 セットの選び方は K 通りあるので、そのすべての場合の正解率を調べ、その平均を、そのモデルの正解率とする。交差検証は、パラメータ選択やモデル選択の基準としてもよく用いられる。この方法の利点は、なるべく多くのデータを教師データとして用いることができつつ、凡化性能が高いモデルを選ぶことができる点である。図（3.12）が交差検証法の仕組みを図示したものである。

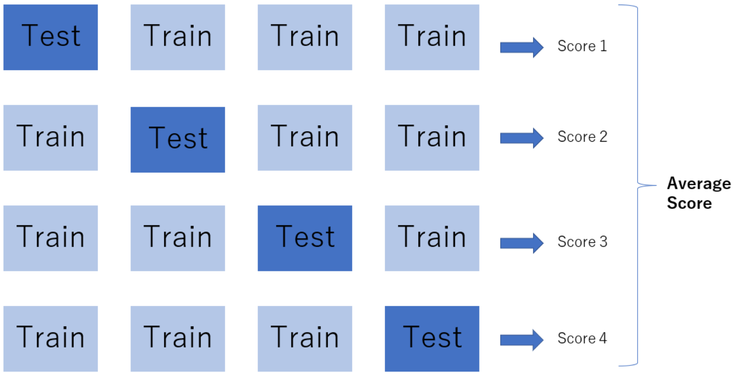

図 3.12. 交差検証の概要図

■グリッドサーチ　グリッドサーチとは、パラメータをチューニングしてモデルの凡化性能を向上させる方法である。モデルに対するパラメータに対し、最良の凡化性能を与える設定を与えるためには、交差検証法を用いて、パラメータのチューニングを行う

まずグリッドサーチでは、データセットをパラメータサーチ用のデータと、教師データ、テストデータに分ける。この時二つのデータを分けずにパラメータサーチをすると、過学習を引き起こし、凡化性能が下がる。そしてパラメータサーチ用のデータに対し、すべてのパラメータの組み合わせで、交差検証を行う。このとき、一番交差検証の結果が良いパラメータを、最良のパラメータとして採用する。そして採用したパラメータで、実際の教師データを学習し、テストデータを予測する。図（3.13）はグリッドサーチでのパラメータ選択とモデル評価の概要を図示したものである。

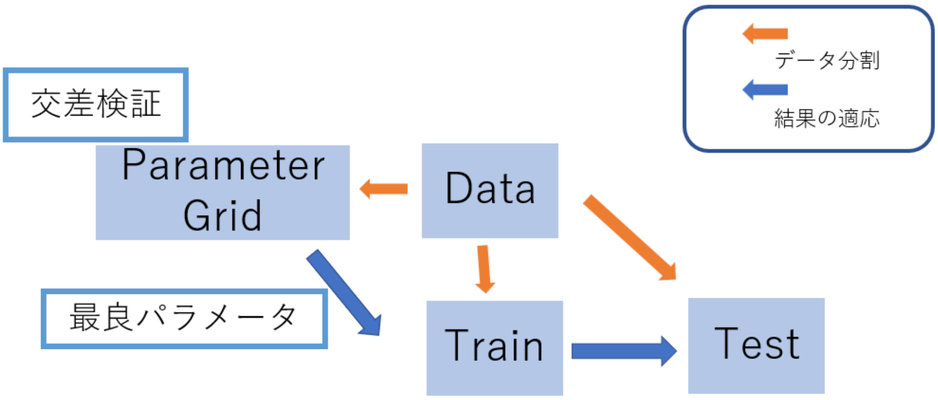

図 3.13. グリッドサーチとモデル選択の概要図

不均一データの均一化

教師あり学習を行う際、教師データが不均一な場合、学習精度が劣化する。ゆえに不均一データを均一にする必要がある。このとき、不均一データを補正するには、少数派クラスのサンプルを増やすか、多数のクラスのサンプルを減らす必要がある。異常検知をする際は、異常サンプルが極端に少なく、正常サンプルを減らして均一化すると、正常・異常両方のサンプルが非常に少なくなる。少ないデータでの学習は、二つのクラス間の違いを学習することが難しくなる。ゆえに、今回は少数派である異常サンプルを増やすことで、不均一データを解消する方法を説明する。以

下では異常サンプルを増やす際に用いた Python のライブラリである、SMOTE(`https://towardsdatascience.com/dealing-with-imbalanced-classes-in-machine-learning-d43d6fa19d2`) について説明する。

■SMOTE　SMOTE とは Python のライブラリの一種であり、不均一データのうち、少数派クラスのサンプルを増やすライブラリである。ただし少数派サンプルを増やす際、既存のサンプルから、新しいサンプルがつくられるので過学習が起きる可能性があることに注意する。以下では SMOTE のアルゴリズムを概略を説明する。(19)

まず既存の少数派クラスのデータを二倍にする。図（3.14）のように少数派クラスのサンプルが存在するとする。このとき、ある少数派クラスのデータ点 A に対し、同クラスの隣接する k 個のデータ $\{A_1, A_2, \ldots, A_k\}$ を用意する。その k 個のデータからランダムに一つ選び、そのデータを A_i とする。そしてその二点 A と A_i の中間地点にある点を新しいデータ点とする。これをすべてのデータ点に対して行う。これにより、少数派クラスのデータ点は二倍になる。この際 n 倍にしたい場合は、k 個の隣接するデータ点に対し、n 個を選び同様な作業を行えばよい。

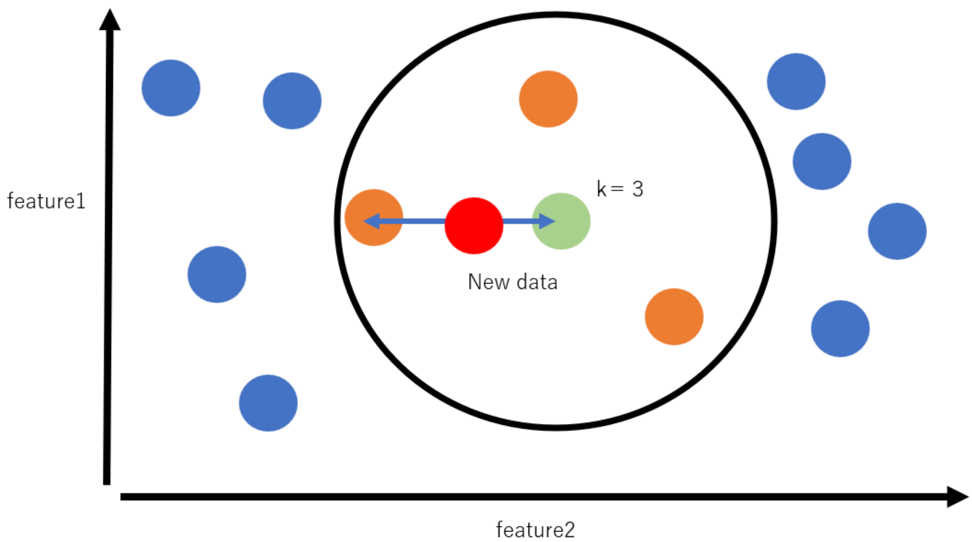

図 3.14. SMOTE の概要

3.3 統計的手法

3.3.1 統計的手法について

統計的手法によるデータ解析とは、与えられたデータの傾向や特性を、データの統計量から、定量的に把握する手法である。本論文では時系列データの解析の際に、統計的手法を用いた。今回はその際に用いた時系列モデルである、自己回帰モデルについて説明する。(16)

3.3.2 自己回帰型モデル

まず $Y_t = \{\boldsymbol{y_1 y_2}, \ldots, \boldsymbol{y_t}\}$ という時系列データがあるとき、この時系列データが自己回帰モデルに従うとする。自己回帰モデルとは、ある時点の出力が過去の出力の線形結合としてあらわわされるとするモデルである。つまり p 次元の自己回帰モデルに従う場合、ある時点 t の観測値 y_t に対し、

$$\boldsymbol{\hat{y}_t} = \boldsymbol{c} + \sum_{i=1}^{p} \phi_i \boldsymbol{y}_{t-i} + \epsilon_t \tag{3.3.2.1}$$

となる。ただし $\phi_i \in \mathbb{R}^{d\times d}$ は $d \times d$ のパラメタ行列であり、ϵ_t は誤差項であり、ホワイトノイズと仮定される。このとき AR モデルによって表されるとなる。ただし $\boldsymbol{\omega} = \sum_{i=1}^{p} \phi_i \boldsymbol{y}_{t-i} + \boldsymbol{c}$ であり、$\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_p, \boldsymbol{c}, \Sigma\}$ とするパラメータの集合である。そして Σ は共分散行列である。この時一番シンプルな AR モデルである、AR(1) モデルを説明する。一次の自己回帰過程を AR(1) 過程といい、時刻 t の値は、時刻 t-1 の値により定まるとする過程である。よって AR(1) 過程は

$$\begin{aligned}\boldsymbol{y_t} &= \phi_1 \boldsymbol{y_{t-1}} + \epsilon_t + c \\ &= \phi_1(\phi_1 \boldsymbol{y}_{t-2} + \epsilon_{t-1}) + \epsilon_t + c \\ &= \phi_1^t \epsilon_0 + \sum_{j=1}^{t} \phi_1^{t-j} \epsilon_j (t = 1, 2, \ldots, T) + c\end{aligned} \tag{3.3.2.2}$$

のようにあらわされる。式 (3.3.2.2) より、$\boldsymbol{y}_0$ から $\boldsymbol{y}_t$ への影響は ϕ_1^t であることがわかる。これより、$\phi_1 < 0$ のとき、過去から現在への影響は時間差が開くほど弱くなることがわかる。このモデルをある時刻 T までの時系列データにあてはめるときは、ϕ_1 と c を求める必要がある。

AR モデルのパラメータを定める際に、用いられるアルゴリズムである SDAR アルゴリズムにより、定められる SDAR モデルを説明する。(20)SDAR アルゴリズムとは、AR モデルのパラメータ推定を、高速に改善したモデルである。通常 AR モデルのパラメータを推定する場合、ユールウォーカ方程式を用いたモーメント法などを用いて行う。しかし、これらはバッチ学習、つまり一度にすべてのデータを一括投入して定めるモデルであり、計算量が多い。一方 SDAR アルゴリズムでは現在の値と、過去に求めたパラメータのみを用いて、現在の値を予想するので、計算量を軽減することができる。加えて定常なデータのみに対応する AR モデルに対し、以下で説明する SDAR モデルでは、過去の統計量の重みを減らす忘却パラメータがあり、これにより非定常な時系列データに対する推定を行える。以下ではまず、SDAR アルゴリズムでのパラメータ推定について説明する。

まず時系列データ Y_t が p 次の AR モデルに従うとする。このとき、$\boldsymbol{y}_t = \boldsymbol{z}_t + c$ とするとき $Z_t = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_t\}$ に対する尤度は

$$L = \prod_{t=1}^{p} p(\boldsymbol{z}_t|\theta) \times \prod_{t=p+1}^{n} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-p}^{t-1} : \theta) \tag{3.3.2.3}$$

となる。よって対数尤度は

$$\log L = \sum_{t=1}^{p} \log p(\boldsymbol{z}_t|\theta) + \sum_{t=p+1}^{n} \log p(\boldsymbol{z}_t|\boldsymbol{z}_{t-p}^{t-1} : \theta) \tag{3.3.2.4}$$

となる。このとき $n >> p$ が成り立つとき、上記の式は、第二項が影響をもつ。今回は簡単のため、$\boldsymbol{z}_t$ が多変数標準正規分布に従うとする。このときの Z_t の確率密度関数は

$$p(\hat{\boldsymbol{z}}_{\boldsymbol{t}}|\boldsymbol{z}_{t-p}^{t-1};\boldsymbol{\theta}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp(\frac{-1}{2}(\hat{\boldsymbol{z}}_{\boldsymbol{t}} - \sum_{i=1}^{p} \phi_i \boldsymbol{z}_{t-i})^T \Sigma^{-1} (\hat{\boldsymbol{z}}_{\boldsymbol{t}} - \sum_{i=1}^{p} \phi_i \boldsymbol{z}_{t-i})) \tag{3.3.2.5}$$

となる。これより以下のように近似できる。

$$\log L \simeq -(n-p)\log((2\pi)^{d/2}|\Sigma|^{1/2}) - \frac{1}{2}\sum_{t=p+1}^{n} \left(\boldsymbol{z}_t - \sum_{i=1}^{p} \phi_i \boldsymbol{z}_{t-i}\right)^T \Sigma^{-1} \left(\boldsymbol{z}_t - \sum_{i=1}^{p} \phi_i \boldsymbol{z}_{t-i}\right) \tag{3.3.2.6}$$

この式を ϕ について偏微分することで、ϕ の尤度推定値が算出される。このときの ω はユール・ウォーカの方程式より、以下の式を満たす。

$$\sum_{i=1}^{p} \phi_i C_{j-i} = C_j (j = 1, 2, \ldots, p) \tag{3.3.2.7}$$

ただしこの時の C_i は

$$C_j = \frac{1}{n-p} \sum_{t=p+1}^{n} \boldsymbol{z}_t \boldsymbol{z}_{t-j}^T = \frac{1}{n-p} \sum_{t=p+1}^{n} (\boldsymbol{y}_t - c)(\boldsymbol{y}_{t-j} - c)^T \tag{3.3.2.8}$$

である。このとき μ の最尤推定量は

$$\hat{\boldsymbol{c}} = \frac{1}{n-p} \sum_{t=p+1}^{n} \boldsymbol{z}_t \tag{3.3.2.9}$$

である。よって式（3.3.2.8）に $\hat{\mu}$ を代入したときの解を $\omega_1, \omega_2, \hat{\ldots}, \omega_p$ とするとき共分散行列 Σ の最尤推定量は

$$\begin{aligned} \hat{\Sigma} =& \frac{1}{n-p} \sum_{t=p+1}^{n} \left(\boldsymbol{z}_t - \sum_{i=1}^{p} \hat{\phi}_i \boldsymbol{z}_{t-i}\right) \left(\boldsymbol{z}_t - \sum_{i=1}^{p} \hat{\phi}_i \boldsymbol{z}_{t-i}\right)^T \\ =& \frac{1}{n-p} \sum_{t=p+1}^{n} \left(\boldsymbol{y}_t - \hat{c} - \sum_{i=1}^{p} \hat{\phi}(\boldsymbol{y}_t - \hat{c})\right) \left(\boldsymbol{y}_t - \hat{c} - \sum_{i=1}^{p} \hat{\phi}_i(\boldsymbol{y}_t - \hat{c})\right)^T \end{aligned} \tag{3.3.2.10}$$

そして x_t の最尤推定量は

$$\hat{y}_t = \sum_{i=1}^{p} \hat{\omega}_i (\boldsymbol{y}_{t-i} - \hat{c}) + \hat{c} \tag{3.3.2.11}$$

となる。これが SDAR アルゴリズムの基本的な流れである。ただし SDAR モデルではさらに推定値に対して、忘却パラメータを設定できる。AR モデルは、時系列過程に定常性を仮定したモデルであるが、変化点検知を行う際は、非定常性を仮定する必要がある。ゆえに、r が小さいほど過去のデータの影響を大きく受けるので、現在の値にフィットする。具体的には、共分散行列 Σ の最尤推定量を更新する際、

$$\hat{\Sigma} := (1-r)\hat{\Sigma} + r(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)^T \tag{3.3.2.12}$$

として更新する。

第 4 章

異常検知アルゴリズムの開発

Contents

4.1 ノイズスペクトルの類別

4.1.1 現象の説明

ノイズスペクトルの類別とは、X 線マイクロカロリメータから得た、8k noise spec が、どのピクセルから検出されたものであるかを、類別する課題である。まず、*Resolve* 装置の X 線マイクロカロリメータは、36 個のピクセルで構成される。ピクセルごとの固有振動数も異なるので、各ピクセルで得られたノイズスペクトルは、ピークが立つ周波数が、ピクセルごとに異なると予想される。図（4.1）は 2022/8/5 20:12:54 に、*Resolve* 装置の地上実験で測定した 8k noise spec のうち、ピクセル 0、ピクセル 1、ピクセル 2 が検出したものをプロットした図である。これを見ると 100Hz 以上の周波数で、いくつかピークが立っていることが確認できる。これらは機械式冷凍機が原因で生じる高調波ノイズである。そして、高調波ノイズのピークの立ち方は、同じ時刻で計測したのにかかわらず、異なる周波数で立っていることが分かる。

図 4.1. 8k noise spec の一例

4.1.2　要件分析

Resolve 装置の地上実験で得られた 8k noise spec に対し、機械学習的手法を用い、ピクセルとノイズスペクトルの関係を学習する。そして、検出元のピクセルが不明な 8k noise spec に対し、学習したアルゴリズムをもとに、どのピクセルから得られたノイズスペクトルであるかを類別する。実際には取得したノイズスペクトルのピクセルが不明な状況は存在しないが、8k noise spec は検出器の健康状態を知る最も有用なデータなので、その扱いに慣れるために設定した課題である。

4.1.3　開発と性能

要件分析に伴い、今回は機械学習的手法を用いて、検出元のピクセルが不明な 8k noise spec が、どのピクセルから検出されたものであるかを類別した。全体の流れとしては、最初に機械学習の精度を上げるため、データセットの前処理をした。次に、8k noise spec の類別を行うため、以下の二つの検証を行った。

①. 教師なし学習であるクラスタリングにより、ノイズスペクトル類別を行った。ここでは、ノイズスペクトルが、どのピクセルが検出したかのラベルを与えなくても、スペクトルの形状から、ピクセルごとに類別できるかを検証した。

②. 教師あり学習である分類により、ノイズスペクトル類別を行った。ここでは、ノイズスペクトルと検出したピクセルのラベルを与え学習させたとき、どの程度の精度で類別できるのかを検証した。

以下では各過程の具体的な作業、そしてクラスタリング、分類によるスペクトル類別の結果を述べる。

データセットの前処理

まず、今回用いたデータセットについて説明する。今回用いたデータセットは、*Resolve* 装置の地上実験で、2021/05/06 03:22:24 から 2022/06/17 00:56:47 までの期間で得られた、8k noise spec のノイズスペクトルである。この期間に得られた 8k noise spec のノイズスペクトルは、3367 ファイルであった。それぞれのファイルには、36 ピクセル分のノイズスペクトルがあるので、全体のサンプル数は $3367 \times 36 = 121212$ sample 存在した。加えて、8k noise spec は、フーリエ変換する際、周波数分解能 1.5Hz ごとの周波数における、パワースペクトルに変換されるので、一つのサンプルは 4096 次元のデータとなった。

次にデータセットの前処理について説明する。実データをそのまま機械学習に用いると、予測の精度が下がるため、データセットの前処理を行った。具体的には、①ノイズスペクトルの常用対数変換、②不要なデータの削除、③次元削除、④標準化を行った。以下順に簡単に説明する。

①今回は各サンプルにおけるパワースペクトルのピークの大きさをなるべく均一にするため、パワースペクトルの大きさは、自然対数変換した。これはデータを視覚化した際に、自然対数変換したグラフのほうが、各サンプルのピークの立ち方の違いが明確に確認できたからである。

②ノイズスペクトルの中身を見ると、いくつかのサンプルはすべての周波数においてパワーが 0 となっていた。これは測定できていないデータであるので、データセットから除いた。加えて一部のデータはパワーの値が 10^{30} を超えていた。これは地上実験の際に、一部測定条件を変えて測定したデータであり、それが通常の測定条件でのデータと混ざっているからである。通常のデータと異なるものを、同じ学習にかけると精度が悪くなるので、これも除いた。これらのデータを取り除いた結果、サンプル数は 121211 sample から 103608 sample に減少した。

③今回は一サンプルあたり 4096 次元であり、高次元のデータであった。次元数が多いと過学習を引き起こしたり、解析に時間がかかるため、PCA を用いて、次元削除を行った。今回は PCA により次元を 4096 次元から、25 次元に減らした。削減する次元数に関しては、次元に対する寄与率のグラフをもとに定めた。各次元に対する、寄与率をプロットした図が図（4.2）である。これより 25 次元の段階で、累計寄与率は 9 割を超し、増加率も滑らかであったので、25 次元まで削減した。

④今回用いた機械学習のアルゴリズムは、データのスケーリングに敏感であった。ゆえに、正規化もしくは標準化を行う必要があった。基本的に正規化を行うと、外れ値の影響を受けやすいので、今回は標準化を行った。

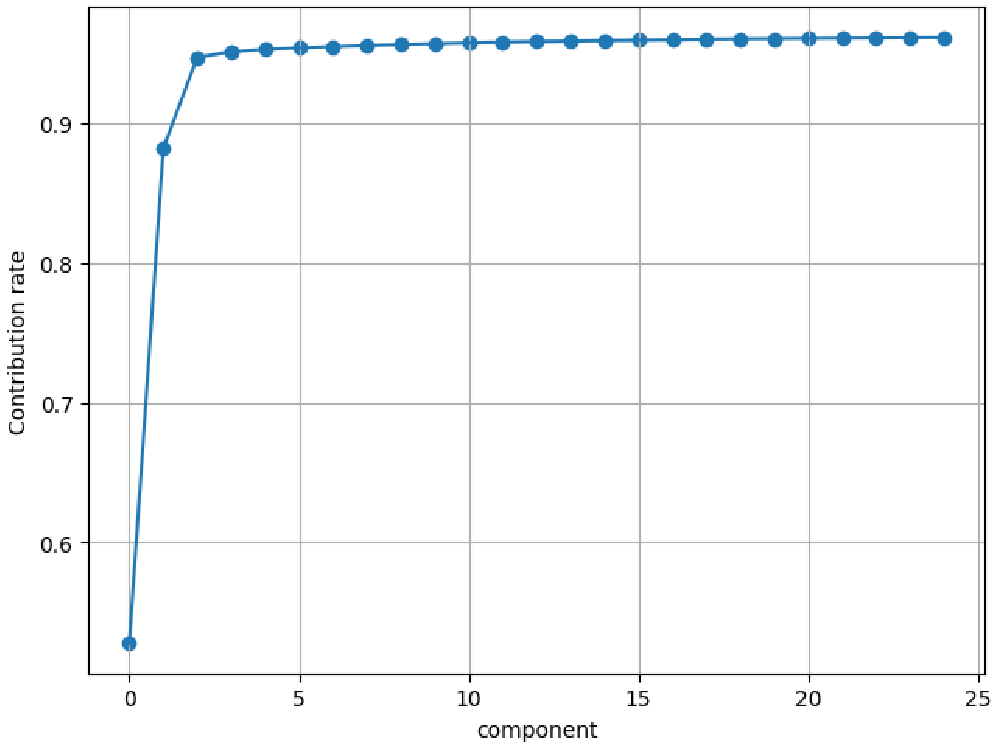

図 4.2. 主成分分析

教師なし学習による類別

データセットの前処理の後、クラスタリングを行い、教師なし学習でピクセルが分類をできるのかを検証した。クラスタリングでは、103608 sample を学習させたので、図（3.6）より、MiniBatchKmeans を用いて行った。

クラスタリングの結果、それぞれのピクセルがどのクラスタに振り分けられたかの個数を記した表が、表（4.1）である。縦軸が実際のピクセル、横軸が振り分けられたクラスタラベルである。ただ、このようなクラスタの振り分け方法であると、各ピクセルがどのクラスタに対応しているのかが、分かりづらい。そこで、各ピクセルごとに、一番多く振り分けられたクラスタを、そのピクセルのクラスタとし、クラスタの名前の再設定をした。例えば、表（4.1）をみると、0 番ピクセルのサンプルは、14 番クラスタに最も多く振り分けられているので、14 番クラスタを 0 番クラスタという名前に再設定した。ただし、一つのクラスタに、複数のピクセルのサンプルが一番多く振り分けられたとき、名前が一つに定まらない。そこで、対応するピクセルのうち、一番小さい数のピクセルを名前として再設定した。このように、名前を再設定したのち、横軸を実際のピクセル、縦軸をそのピクセルのサンプルが、一番多く振り分け割れたクラスタとして図示したもが、図（4.3）である。また、それぞれのピクセルに属するサンプルの内、図（4.3）で対応するクラスタへ、実際に振り分けられたサンプルの割合をプロットした図が、図 (4.4) である。まず図（4.3）を見ると、すべてのピクセルとクラスタは、一対一対応になってはいないことが分かった。ゆえに、教師なし学習のみでは、ノイズデータからピクセル分類を完全に行うことが出来ないことがわかる。加えて図（4.4）をみると、約半分のクラスタは正解率が 0.9 を越しているが、残りのものは 0.5-0.9 の間となり、正解率が低いことがわかる。このとき、なぜ一部のピクセルが、同じクラスタに振り分けられたのか、分析は教師あり学習の結果をもとに行う。

ここでこの結果より、教師なし学習でのクラスタリングの際、なぜ一部のピクセルが、同じクラスタに振り分けられたのか、分析する。例えばクラスタリングの結果である図（4.3 を見ると、3 番クラスタに、8 番ピクセルや 18 番ピクセルのサンプルも、最も多く振り分けられていることがわかる。また今回の結果を見ると、3 番クラスタ、8 番クラスタ、18 番クラスタが全体的に多くのピクセルに振り分けられていることがわかる。つまり 3 番クラスタや 8 番クラスタのスペクトルは、ほかのクラスタの特徴が混在しているスペクトルが多いことがわかる。ゆえに、クラスタリングを行う際、特定のクラスタに分類されなかったと考えられる。この特徴は、次の 4.1.3 でも見られた。

実際のピクセル

判別されたクラスタ

0	13	0	0	0	0	0	0	0	104	0	0	0	0	2868	0	0	0	0	0	0	9	0	0	0	2	0	0	0	0	0	0	0	0	11	0
0	13	0	0	0	0	0	0	0	108	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	2586	0	0	0	0	0	0	0	0	88	0
0	13	0	0	0	0	0	0	0	105	0	0	0	0	0	0	2887	0	0	0	0	7	0	0	0	0	0	0	0	0	0	1	0	0	11	0
0	12	1	0	0	0	0	0	0	102	0	0	0	0	0	0	11	0	0	0	0	9	0	0	0	9	0	0	0	5	0	0	0	0	2063	0
0	12	0	0	0	0	0	0	0	41	2750	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	7	0
0	12	5	0	0	0	0	0	0	111	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	2543	0	0	0	0	143	0
0	12	0	0	0	0	0	0	0	10	0	0	2765	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	11	0
0	12	0	0	0	0	0	0	0	55	0	0	0	0	6	0	0	0	0	2567	0	7	0	0	0	0	0	0	3	0	0	0	0	0	203	0
0	13	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	9	0	0	0	0	2698	0
0	13	2	0	0	0	0	0	0	101	0	0	0	2840	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	94	0
0	13	2	0	0	0	0	0	0	90	0	0	0	1	0	0	0	0	0	0	0	7	2941	0	0	0	0	0	0	0	0	0	0	0	10	0
0	10	3	0	0	0	0	0	0	35	0	0	0	0	0	3	2	0	0	0	2102	8	0	0	0	0	0	973	0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0	0	1	0	0	0	0	0	1521	0	0	0	0	0	7	0	0	0	0	0	1	0	0	0	0	0	0	6	0
0	13	902	0	0	0	0	0	0	97	0	0	0	0	0	0	0	2098	0	0	0	8	0	0	0	4	0	4	0	0	0	0	0	0	25	0
0	13	1	0	0	0	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	2933	0	0	0	0	0	13	0
0	11	1	0	0	0	0	0	2963	55	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	2	0	0	12	0
0	13	0	0	0	0	0	0	0	105	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	2932	0	0	13	0
0	11	2	0	3339	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	8	0	0	0	0	0	0	9	0
0	13	2	0	0	0	0	0	0	117	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	2896	0
0	13	2713	0	0	0	0	0	0	116	0	0	0	0	0	0	2	0	0	0	0	7	0	0	0	0	0	12	0	0	0	0	0	0	0	0
0	13	3	0	0	0	0	2249	0	118	0	0	0	0	0	0	473	0	0	0	0	7	0	0	0	0	0	2	0	0	0	0	0	0	8	0
0	13	2	0	0	0	0	0	0	115	0	0	0	0	0	0	0	0	0	0	0	7	0	0	2890	0	0	0	0	0	0	0	0	0	39	0
0	13	2	0	0	0	0	0	0	88	0	2752	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	9	0
0	13	6	0	0	0	0	0	0	111	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	84	0	0	0	0	0	2669	4	0
0	13	2	2776	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	8	0
918	13	49	0	0	0	0	0	0	118	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	12	0	807	0	11	0	0	958	0
2779	12	1	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	13	0	0	0	0	0	0	6	0
0	13	2	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	2756	0	0	7	0	0	1	0	0	2	0	0	0	0	0	0	11	0
0	13	0	0	0	0	0	0	0	122	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	58	0	0	0	5	0	0	0	2710	0
0	13	4	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	1	0	0	0	1119	0	0	0	6	1659
0	13	2	0	0	0	0	0	0	121	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	2702	0	0	0	38	0	0	0	10	25
0	13	2	0	0	2836	0	0	0	32	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	9	0
0	13	2902	0	0	0	0	0	0	122	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	1	458	0
0	13	503	0	0	0	0	0	0	122	0	0	0	0	0	0	0	0	0	0	0	7	0	2042	0	0	0	0	0	0	0	0	0	0	198	0
0	12	3	0	0	0	2962	0	0	40	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	9	0
0	13	2	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	2788	0	9	0

表 4.1. 検出したピクセルと振り分けられたクラスタの表

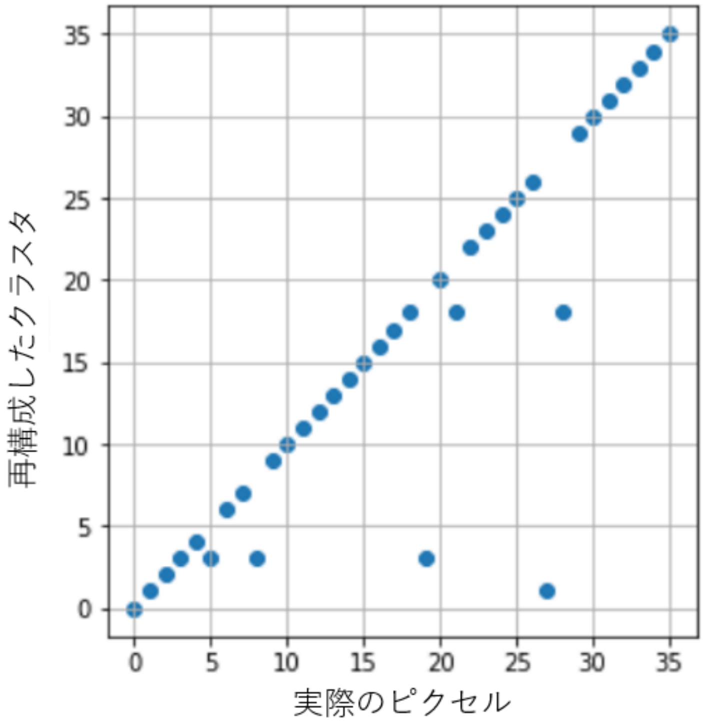

図 4.3. 検出したピクセルに対する一番多く振り分けられたクラスタの関係

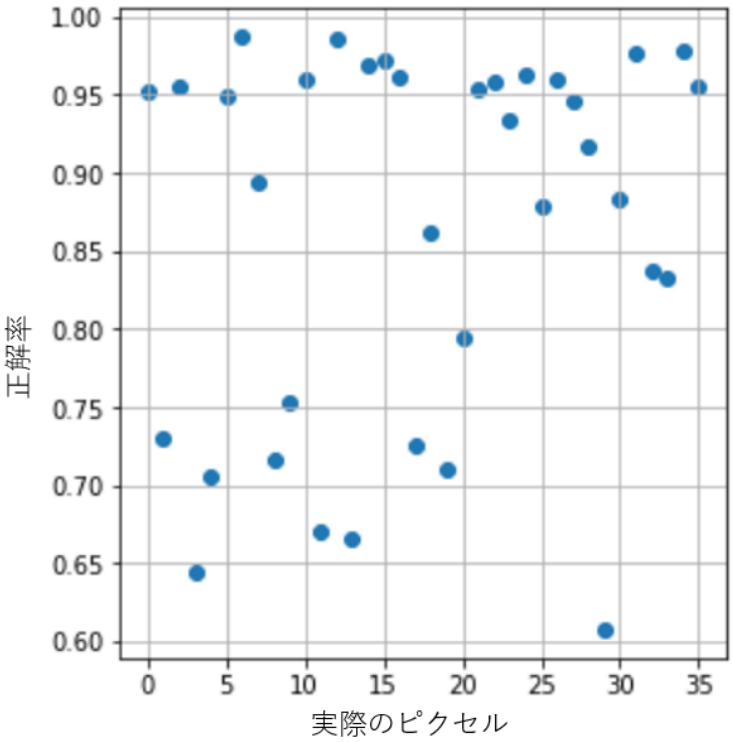

図 4.4. クラスタリングにおけるピクセル類別の正答率

教師あり学習による類別

教師なし学習の結果、ピクセル類別ができなかったので、分類によるピクセル類別を行った。ここでは教師データとして、各ノイズスペクトルと対応するピクセルのラベルを、一緒に学習させた。教師あり学習では、データセットをパラメータチューニング用、交差検証用、テスト用の三種類に分けた。データセットは全部で 103611 sample あるため、それぞれのデータセットの大きさは 36263 sample、31082 sample、36263 sample となった。まず、分類におけるモデルを選ぶ際、テストサンプルが 36263 sample であることから、図 (3.6) を見ると、LinearSVC が良いことがわかった。しかしデータセットが多い場合は、100000 sample を超えていなくても、SGDClassifier も有効であるため、SGDClassifier も同様にモデルとして検討し、どちらが良いか比較した。

初めに、モデルのパラメータを定めるため、パラメータチューニングを行った。今回はそれぞれのモデルから、scikit-learn の UserGuide を参照しながら、重要なパラメータを検証した。まず、LinearSVC の主要なパラメータは表（3.2）となる。このパラメータのうち、重要なパラメータは、loss パラメータと C パラメータ、penalty パラメータであった。故にこの 3 つのパラメータを、3 分割の交差検証用いたパラメータ探索をした。その結果 loss パラメータは squared_hinge、C パラメータは 1、penalty パラメータは l2 に設定した。次に、SGDClassifier の主要なパラメータは表（3.3）となる。このなかで scikit-learn の UserGuide より、max_inter パラメータは 14 に設定した。そのほかのパラメータで重要なパラメータは、alpha パラメータ、loss パラメータ、penalty パラメータであった。故に、これらのパラメータを、3 分割の交差検証を用いた GridSerach で探索した。その結果、alpha を 0.000001、loss を squared_hinge、penalty パラメータは l2 に設定した。

次に、LinearSVC と SGDClassifie のどちらがモデルとして適しているか、交差検証を行い比較した。交差検証の分割数は 5 回とした。5 回の平均はそれぞれ 0.962、0.977 となった。これより SGDClassifie をモデルとして採用した。

最後に SGDClassifier を用いて、テストデータの分類を行った。まず、それぞれのピクセルと分類されたカテゴリに対する、振り分けられたサンプルの数の表が図 (4.2) である。これを heatmap に変換したものは、図（4.5）となった。そして、各ピクセルの振り分けられたカテゴリに対する正答率は、図（4.6）となった。図（4.6）を見ると、二つのピクセルを除き、すべてのピクセルで正答率が 0.95 以上となっていることが分かり、高水準で予測ができていることがわかる。

実際のピクセル

判別されたクラスタ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	868	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	797	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	834	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	758	0	1	0	0	5	0	0	0	0	0	0	0	2	0
4	0	0	0	15	821	0	0	0	4	0	0	0	1	0	0	0	0	0
5	0	0	0	3	0	792	0	0	4	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	820	0	0	0	0	0	0	0	0	0	2	0
7	0	0	0	2	0	0	0	802	3	0	0	0	0	0	0	0	0	0
8	0	0	0	5	0	0	0	0	798	0	0	0	0	0	0	0	0	0
9	0	0	0	3	0	0	0	0	0	864	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	875	0	0	0	0	0	1	0
11	0	0	1	2	0	0	0	0	1	0	0	921	0	0	1	0	0	0
12	0	0	0	3	0	0	0	0	1	0	0	0	442	0	0	0	0	0
13	0	0	0	3	0	0	0	0	6	0	0	0	0	911	0	0	1	0
14	0	0	0	2	0	0	0	0	6	0	0	0	0	0	906	0	0	0
15	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	944	0	0
16	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	946	0
17	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	1	897
18	0	0	0	3	0	0	0	0	3	0	0	0	0	0	1	0	1	0
19	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
20	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	1	0
23	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
25	0	0	0	2	0	0	0	0	4	0	0	0	0	0	0	0	1	0
26	0	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	0	0
27	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0
28	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0
29	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	2	0	0	0	0	5	0	0	0	0	0	0	0	0	0
31	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0
32	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	2	0
34	0	0	0	1	0	0	0	0	7	0	0	0	0	0	0	0	0	0
35	0	0	0	5	0	0	0	0	4	0	0	0	0	0	1	0	0	0

	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
0	1	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
2	4	0	0	0	0	0	0	4	0	0	4	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	118	0	0	0	1	0	0	0
4	2	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
5	8	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0
7	3	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
8	15	0	0	0	0	0	0	0	0	0	2	0	0	0	4	0	0	0
9	12	0	0	0	0	0	0	0	0	0	5	0	0	0	1	0	0	0
10	16	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
11	2	0	0	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0
12	1	0	0	0	0	0	0	3	0	0	4	0	0	0	0	0	0	0
13	3	0	0	0	0	0	0	0	0	0	9	0	0	0	1	1	0	0
14	3	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
15	7	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0
16	2	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
17	2	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
18	852	0	0	0	0	0	1	7	0	0	25	0	0	0	6	0	0	0
19	1	827	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	2	0	818	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0
21	13	0	0	840	0	0	0	0	0	0	1	0	1	0	1	0	0	0
22	3	0	0	0	886	0	0	0	0	0	12	0	0	0	0	0	0	0
23	7	0	0	0	0	863	0	1	0	0	8	0	0	0	1	0	0	0
24	1	0	0	0	0	0	807	6	0	0	2	0	0	0	0	0	0	0
25	5	0	0	0	0	0	0	862	0	0	1	0	0	0	1	0	1	0
26	0	0	0	0	0	0	0	16	833	0	7	0	0	0	0	0	0	0
27	1	0	0	0	0	0	0	1	0	804	18	0	0	0	0	0	0	0
28	8	0	0	0	0	0	0	0	0	0	832	0	0	0	0	0	0	0
29	2	0	0	0	0	0	0	0	0	0	31	822	1	0	0	0	0	0
30	6	0	0	0	0	0	0	0	0	0	18	1	833	0	0	0	0	0
31	6	0	0	0	0	0	0	0	0	1	2	0	0	873	0	0	0	0
32	14	0	0	0	0	0	0	0	0	0	12	0	0	0	924	0	0	0
33	1	0	0	0	0	0	0	0	0	0	9	0	0	0	0	825	0	0
34	1	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	891	0
35	7	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	869

表 4.2. 検出したピクセルと分類されたカテゴリに対するサンプル数

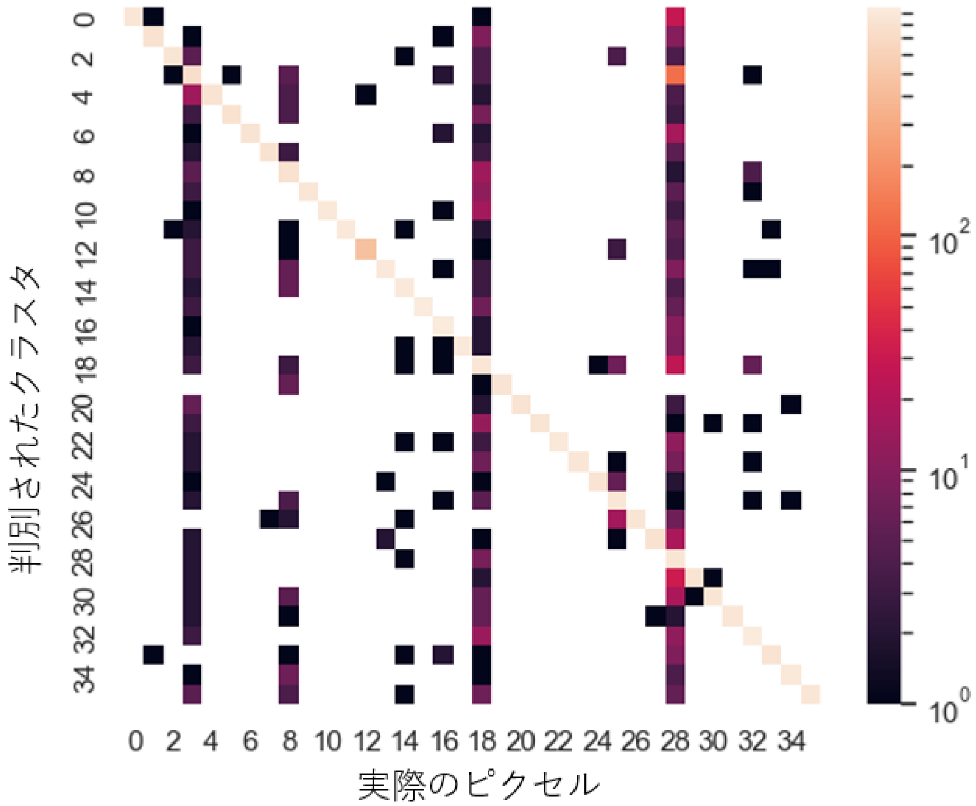

図 4.5. 検出したピクセルと分類されたカテゴリに対するサンプル数のヒートマップ

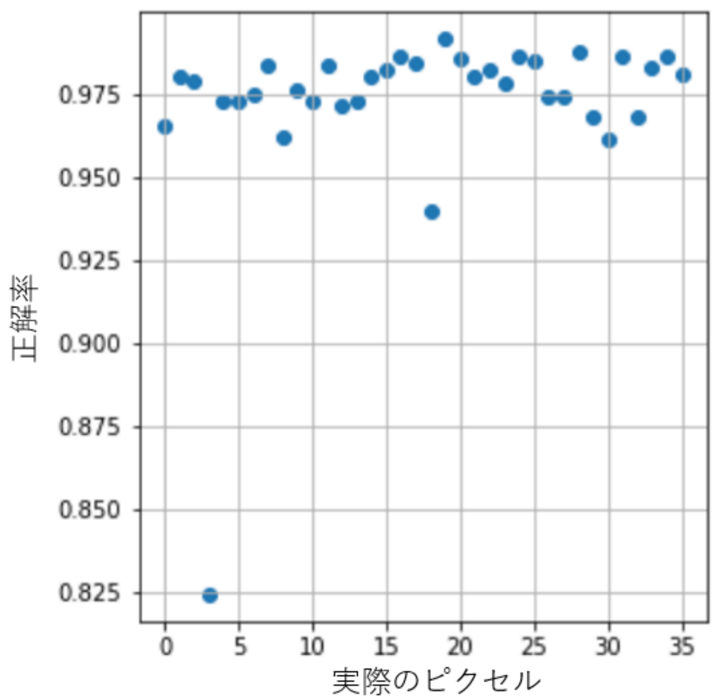

図 4.6. 分類におけるピクセル類別の正答率

4.2 ノイズスペクトルの異常

4.2.1 現象の説明

Resolve 装置で X 線を検出する際、温度依存する抵抗値の変化は図 4.7 のようになる。また温度依存する抵抗値の変化のスペクトルは、図 4.8 となる。図 4.8 を見ると、X 線イベントのエネルギー分解能に影響するのは、350 Hz 以下である。ゆえにこの周波数帯のノイズスペクトルの安定化は、*Resolve* 装置の X 線分解能を劣化させないために必要である。

350 Hz 以下の間には、主に以下の四つの原因でノイズスペクトルにピークが生じる。

（1）機械式冷凍機によるピーク。*Resolve* 装置には、Stirling 冷凍機と Joul-Thomson 冷凍機の、二つの機械式冷凍機が搭載されている。それらの機械式冷凍機の振動により、それぞれ 15 Hz とその倍波、52 Hz とその倍波でのピークが、ノイズスペクトルに生じる。このとき倍波でのピークが生じるのは、正弦波からずれているためである。(A.5)

（2）リアクションホイール (RW) よるピーク。リアクションホイールとは、XRISM 衛星に搭載される、姿勢制御系のサブシステムの一つである。リアクションホイールの振動により、25 Hz から 75 Hz の間でピークが、ノイズスペクトルに生じる。

（3）MTQ ノイズ。磁気トルカ (MTQ) も、XRISM 衛星に搭載される、姿勢制御系のサブシステムの一つである。MTQ はコイルに電流をかけて地球磁場と作用させることで、XRISM 衛星に角運動量を与える。この時 127 Hz のパルス幅変調で電流を調節するので、127 Hz とその倍波でのピークが、ノイズスペクトルに生じる。図 4.9 は MTQ ノイズの一例である。MTQ ノイズはピクセル 0、9、18、27 から検出されたノイズスペクトルはピークが立ちやすく、それ以外のピクセルから検出されたノイズスペクトルだとピークが出にくい。これはピクセルのワイヤーの順番が影響している。検出器からデータを PSP に送信する際、ピクセル 0、9、18、27 のワイヤーが外側にあり、影響を受けやすいからと考えられる。(21)

（4）機械式冷凍機による Beat ノイズ。10 Hz 以下にピークが生じるノイズスペクトルは、機械式冷凍機の共鳴による、Beat ノイズが原因と考えられる。(22) 図 4.10 が機械式冷凍機による Beat ノイズの一例である。Beat ノイズとは、異なる二つの周波数の波が干渉する際に、生じるノイズである。例えば周波数 f_1 の波と、周波数 f_2 の波が干渉すると、

$$\cos(2\pi f_1 t) + \cos(2\pi f_2 t) = 2\cos\left(2\pi\frac{f_1 - f_2}{2}t\right)\cos\left(2\pi\frac{f_1 + f_2}{2}t\right) \tag{4.2.1.1}$$

という波となる。ここで、$\frac{f_1 - f_2}{2}$ が Beat ノイズの周波数となる。通常二つの波が合成された場合、フーリエ変換すると元の二つの波に分解される。しかし合成された波に、ある種の非線形効果が働く場合、Beat ノイズの周波数もピークが生じてしまう。

この四つのピークに対し一方、機械式冷凍機による Beat ノイズは、機械式冷凍機が特定の組み合わせになる場合に出現する。MTQ ノイズは MTQ の駆動中、特定のパラメータの場合にピークの高さが高くなる。そこで今回は、Beat ノイズと MTQ ノイズを異常なノイズとみなし、今回の検出対象とした。

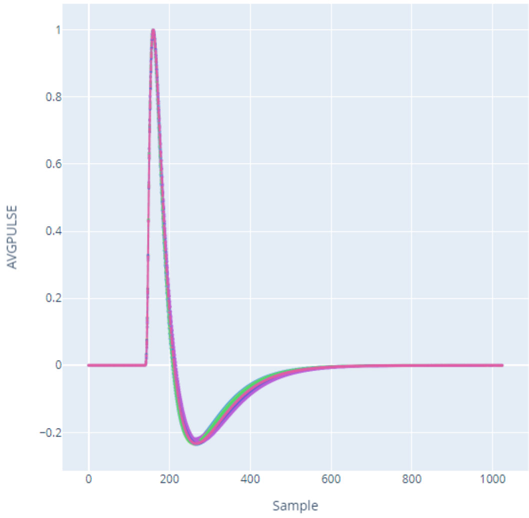

図 4.7. X 線入射時の温度変化

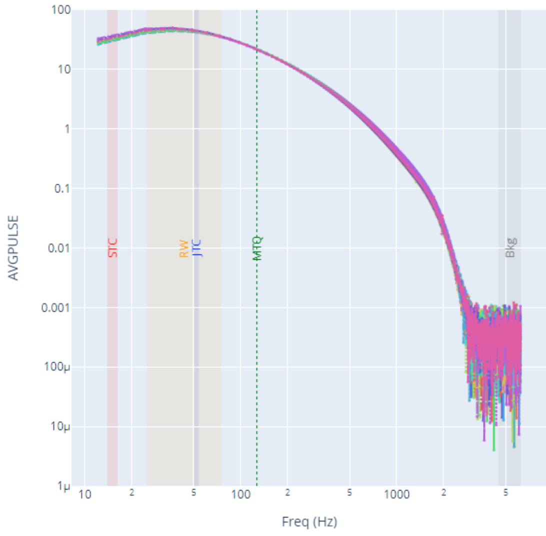

図 4.8. X 線入射時のスペクトル

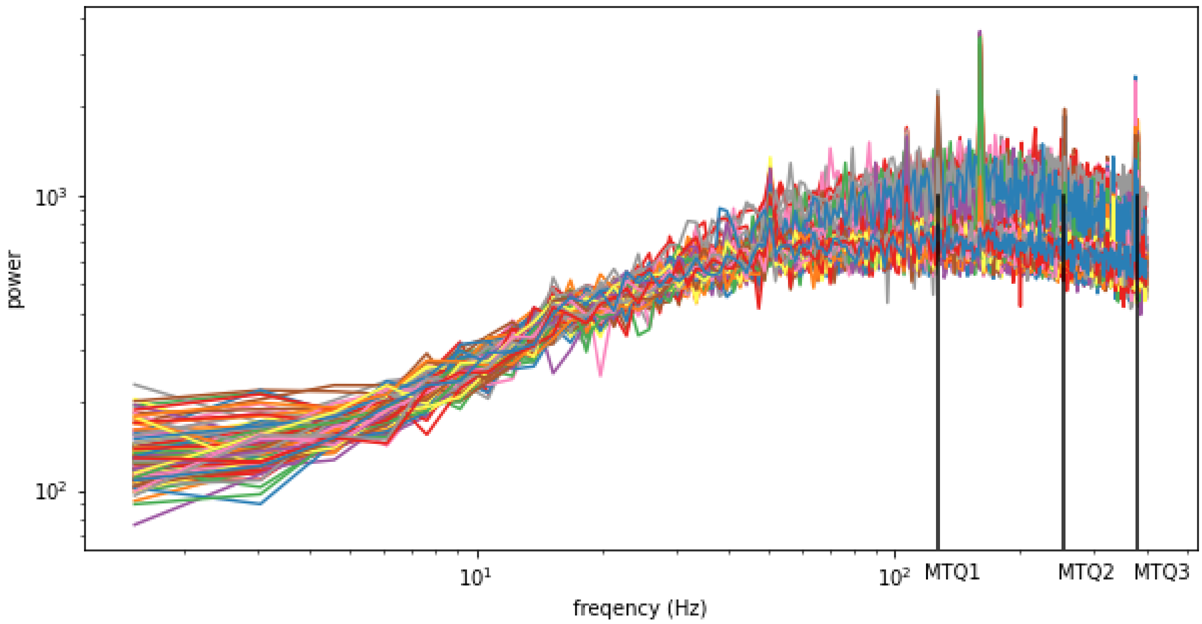

図 4.9. MTQ ノイズの一例

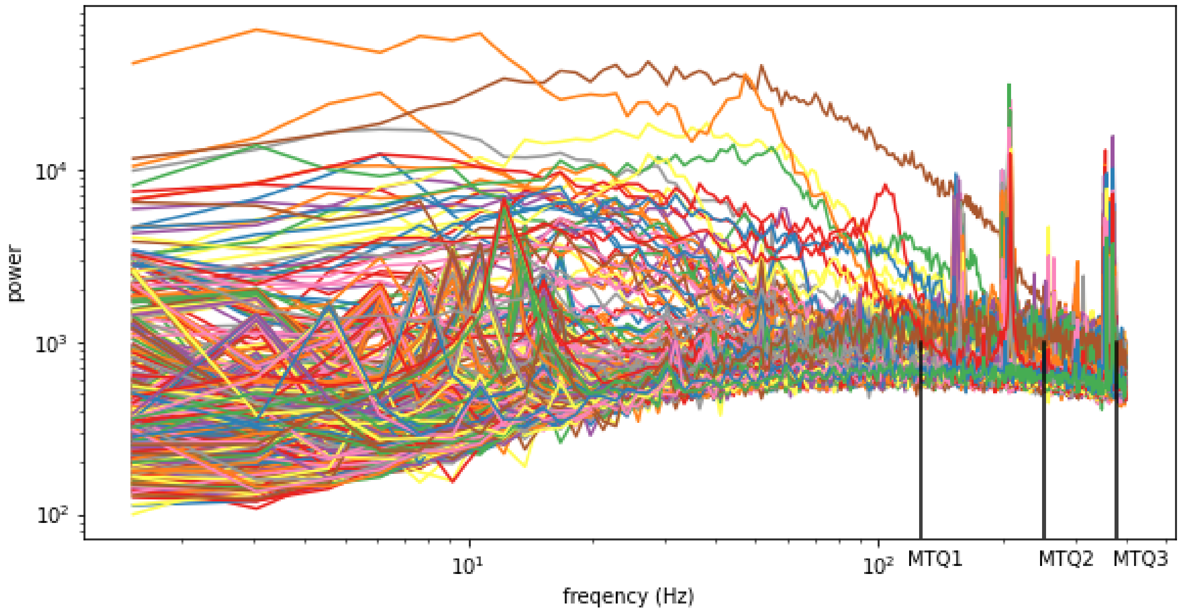

図 4.10. Beat ノイズの一例

4.2.2　要件分析

Resolve 装置の地上実験で得られた 8k noise spec に対し、機械学習的手法を用い、異常なピークが立っているスペクトルを検出する。ただし対象とする異常ノイズスペクトルは、Beat ノイズと MTQ ノイズが生じているノイズスペクトルとする。

4.2.3　開発と性能

要件分析に伴い、今回は機械学習的手法を用いて、Beat ノイズと MTQ ノイズを検出する検出器の作成を試みた。

全体の流れとしては、最初に機械学習の精度を上げるため、§4.1 と同様に、データセットの前処理を行った。そして、前処理を行ったデータセットを用いて、以下の四つの検証を行った。

①. MTQ ノイズと Beat ノイズが同時に検出できるかを試みた。これは、二つが同時に検出できれば、実装する際にシンプルなコードとなるので検証した。同時検出をするアルゴリズムを選ぶにあたり、初めに教師あり学習を用いた検出を検証した。

②. ①と同様 MTQ ノイズと Beat ノイズが同時に検出できるかを試み、Kmeans を用いた再構成誤差をによる検出を検証した。しかし結果より、MTQ ノイズ、Beat ノイズを別々に検出することも検証するべきであると考え、それぞれに対し、別の検出器を作成する③と④の工程を行った。

③. MTQ ノイズに対して、教師あり学習を用いた検出の検証を行った。

④. Beat ノイズに対して、再構成誤差による異常検知を利用した検出の検証を行った。

以下では各過程の具体的な作業、それぞれの手法による異常ノイズスペクトル検出の結果を述べる。

データセットの前処理

まず、今回用いた三つのデータセットについて説明する。

一つ目は MTQ ノイズのデータセットである。これは、2021/9/14、2021/9/15 に行われた XRISM 衛星のサブ

システム試験、2022/6/9、2022/6/11 に行われた XRISM 衛星のシステム試験で得られた、MTQ が駆動しているときの 8k noise spec のデータセットである。この期間で得られた MTQ ノイズのファイルは、サブシステム試験時に 12 ファイル、システム試験時に 5 ファイルである。各ファイルには 36 ピクセル分のノイズスペクトルがあるので、サンプル数は $17 \times 36 = 612$ sample であった。

二つ目は Beat ノイズのデータセットである。これは、2019/12/6、2019/12/11 に行われた XRISM 衛星の試験で得られた、Beat ノイズが発生した 8k noise spec のデータセットである。この期間で得られた Beat ノイズのファイルは 40 ファイルであった。よって、サンプル数は $40 \times 36 = 1440$ sample であった。

三つ目は正常なデータセットである。これは XRISM の地上実験で、2021/05/06 から 2022/06/17 までで得られた、8k noise spec のデータセットである。この期間に得られたデータファイルから、MTQ ノイズのデータセットと重複したデータを除くと、3356 ファイルとなった。よって、サンプル数は $3356 \times 36 = 120816$ sample であった。

次にデータセットの前処理を行った。具体的には 4.1 と同様に、不要なデータの削除を行った。その結果、MTQ ノイズデータセットは 564 sample、Beat ノイズデータセットは 1356 sample、正常なデータセットは 103239 sample となった。加えて MTQ ノイズのデータセットや Beat ノイズのデータセットは 2021/05/06 から 2022/06/17 までの、すべての MTQ ノイズや Beat ノイズを網羅していない。つまり正常データセットには、MTQ ノイズと Beat ノイズが含まれている可能性があった。なので各検証の前に、必要に応じて、127 Hz、254 Hz、381 Hz、10 Hz 以下で、ノイズスペクトルのパワーが 1500 以上となっているものを取り除た。これによりある程度の MTQ ノイズと Beat ノイズを正常データセットから取り除いた。

教師あり学習を用いた MTQ ノイズと Beat ノイズの検出

データセットの前処理の後、MTQ ノイズと Beat ノイズを同時に検出できるかを検証した。このとき用いる異常検知アルゴリズムとして、初めに教師あり学習の SGDClassifier を用いて学習を行った。まず必要なサンプルは、GridSearch によるパラメータ探知用のサンプル、学習用サンプル、テスト用サンプルである。そこで GridSearch 用サンプルは、51619 sample を正常スペクトル、282+678 sample を MTQ ノイズと Beat ノイズ合わせた異常サンプルとした。次に学習用サンプルは、51620 sample を正常スペクトル、282+678 sample を MTQ ノイズと Beat ノイズ合わせた異常サンプルとして学習させた。最後にテスト用サンプルは、正常データセットから 1032 sample を抽出し、そのうち明らかに正常であるサンプルを目で確認しながら、876 sample 抽出した。それに加え、164 sample の MTQ ノイズと 136 sample の Beat ノイズを教師データの異常サンプルとして抽出した。

各サンプルを正規化させた後、GridSearch を行った。SGDClassifier の主要なパラメータは表（3.3）である。パラメータ探索の結果パラメータは、alpha が 1e-05、loss が squared_hinge、penalty が l2 となった。

パラメータ探索の結果より、最適なパラメータを用い、SGDClassifier を行った。この結果、混合行列は表（4.3）となった。これより、異常データの多くが正常データと判断され、精度が非常に悪くなった。これについて、用いた正常データセットに、原因が二つあると考えた。ただし、この時用いた教師の正常データセットのノイズスペクトルの一部が、図（4.11）である。

表 4.3. 正常データセット無処理での教師あり学習の混合行列

		判断された結果		
		Beat ノイズ	MTQ ノイズ	正常
実際の結果	Beat ノイズ	9	0	669
	MTQ ノイズ	0	0	164
	正常	0	0	510

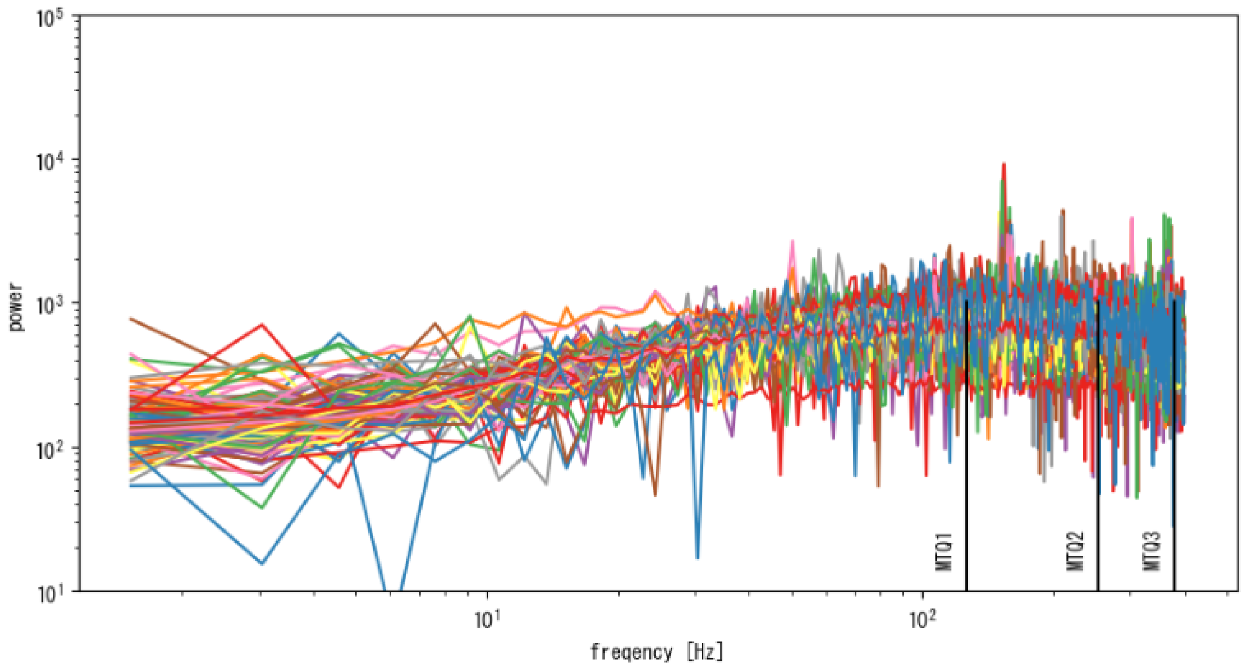

図 4.11. 学習させた正常サンプル

一つ目が、正常データセットに異常ノイズスペクトルが含まれていることである。これに関して、データセットの前処理の段階で、1500 を閾値とし、ある程度正常データセットから MTQ ノイズと Beat ノイズを除いた。しかし 127 Hz とその倍波で 1500 以下のパワーを持つ MTQ ノイズ、10 Hz 以下で 1500 以下のパワーを持つ Beat ノイズも存在する。これが正常データセットにも含まれており、正常データセットに含まれる低いピークの MTQ ノイズと Beat ノイズが、学習結果に影響を与えていた可能性があった。

二つ目が、正常データセットに機械式冷凍機によるノイズスペクトルへのピークが、高いものが含まれていることである。機械式冷凍機によるピークは、冷凍機が常に駆動していることから、常に生じている。しかしピークの高さは、ばらつきが大きいことが図（4.11）からもわかる。ゆえに機械式冷凍機によるピークの高さが、正常、異常のラベル付けに対する判断基準に、影響を与えている可能性があった。

これらの理由より、結果を改善するためには、正常データセットのピークを滑らかにする必要があると考えた。これは、正常データセットに含まれる MTQ ノイズや Beat ノイズのピークを低くすることで、正常データセットに含まれる異常ノイズスペクトルの影響を減らす。それと同時に、機械式冷凍機のピークを下げることで 127 Hz とその倍波、10 Hz 以下のピークの高さの、ラベル付けに対する影響力を上げるためである。そこで正常データセットのいくつかのサンプルをランダムにとり、そのサンプルの平均スペクトルを、正常サンプルとして用いることにした。これにより、異常に高くピークが立っているものが平均化され、正常データセットのスペクトルが、平坦になることが予想された。この作業を本論文では以後平坦化と呼び、一回の平均を取る際に用いるサンプル数を平均化するサンプル数と呼ぶ。このとき平均化するサンプル数を大きくすると、より滑らかになり、正常であるのに、機械式冷凍機のピークが高く異常と判断されるサンプルも多くなることが予想された。ゆえに平均化するサンプル数を、どの値にするべきかも検証する必要が出た。よって、データセットを図（4.12）のように分け、SGDClassifier のパラメータ、平均化するサンプル数の最適な値を検証し、テストデータを適用した。ただしこのとき以下のような流れで検証をした。以下ではそれぞれの手順について補足する。

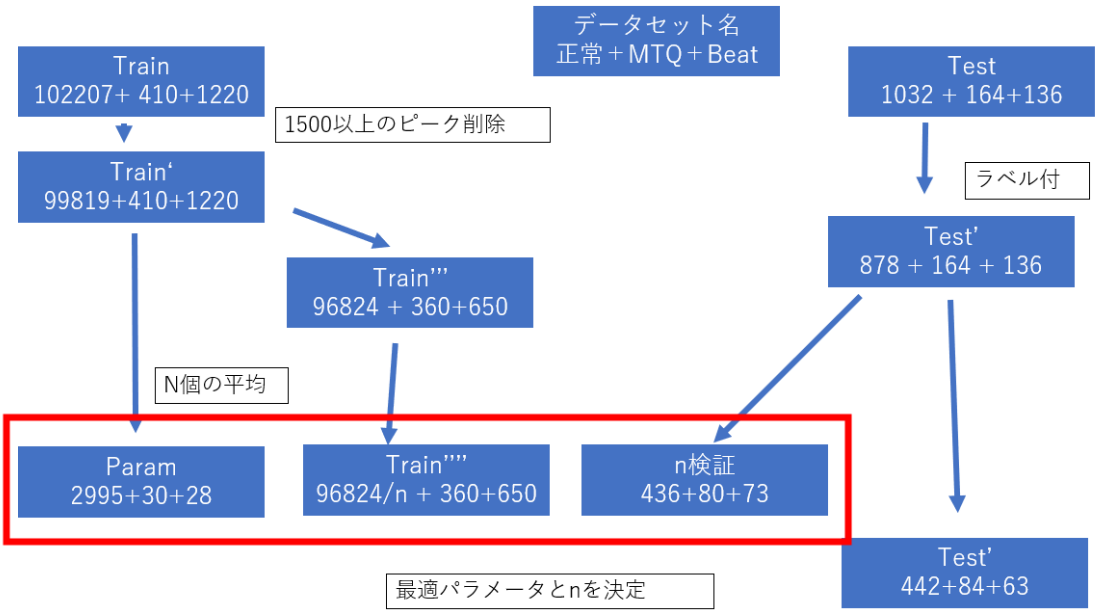

図 4.12. 教師あり学習を用いた同時検出のサンプル

Step 1 Train データをパラメータ探索用データ (Param) と学習用データ (Train’) に分け、Test データセットを平均化するサンプル数検証用データ (n 検証) とテストデータ (Test’) に分けた。

Step 2 n=10 個のデータを Train’ からランダムに抽出し、それの平均を Train” に追加した。これをすべてのサンプルが Train’ からなくなるまで、繰り返した．ただし残りの Test’ データが n 未満になった段階で終了した。

Step 3 Train” のデータセットを SGD Classifier に学習させ、２分割した Param のデータセットに対してテストした。このとき二つの F 値の平均を各パラメータに対して比較し、最も F 値のよいパラメータを最良のパラメータを検証した。

Step 4 Step3 で得られたパラメータによる SGD Classifier に Train” のデータセットを学習させ、n 検証データセットを用い、F 値を求めた。

Step 5 n の値を 10 増やし、n が 200 になるまで step2 から step4 を繰り返した。そして各 n に対する F 値を比較し、一番高いものを最適な n とした。

Step 6 step5 で得られた n と、n に対するパラメータを Train” に用いて学習させ、Test’ に対しテストを行い、正常・異常のラベル付をした。

Step1 では、図（4.12）のようにサンプルを分けた。Step2 では教師データのうち、正常なデータを平坦化にした。そして Step3 では、平坦化データを用いて最良のパラメータを検証した。このとき通常であれば、GridSearch を用いて検証を行うのだが、今回の教師データの正常なデータは、平坦化したのでテストすべき正常なデータとは異なる。なので平坦化したデータに対し交差検証を行っても意味がないので、テストデータに加工前のデータを用いてテストを行い、パラメータを検証した。

Step4、Step5 では平坦化する際の、平均化するサンプル数の検証を行った。そして今回は異常検知なので、F 値をパラメータを定める基準とした。Step5 での平均化するサンプル数を 10 から 200 まで 10 づつ変化させ、そのときの適合率、再現率、F 値をプロットした図が図（4.13）である。これを見ると、予想通り平均化するサンプル数を

上げると適合率が上がるが、再現率が下がることがわかる。これより、F 値が最大である 30 個を平均化するサンプル数として用いた。

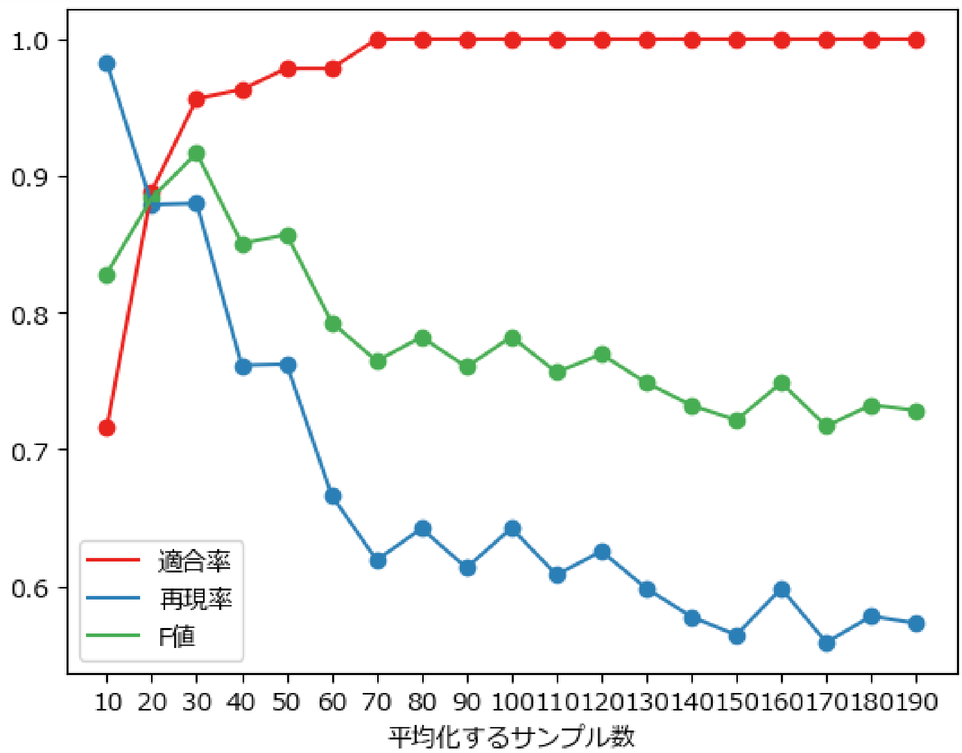

図 4.13. 各平均化サンプル数に対する陽性率と偽陽性率、F 値

30 サンプルを平均化するサンプル数とし、それに対し最適なパラメータである alpha を 1e-06、loss を squared_hinge、penalty を l2 での SGDClassifier をもちいて、テストデータを予想した結果は、表（4.4）となった。平均を取らない場合と比較すると、再現率は 0.75 以上となり、異常の検知精度は向上した。ただ更なる再現率の向上のため、異常なノイズスペクトルであるのに、正常と判断されたノイズスペクトルが、なぜ正常と判断されたのかを分析する。

表 4.4. 教師あり学習の混合行列

		判断された結果		
		Beat ノイズ	MTQ ノイズ	正常
実際の結果	Beat ノイズ	55	2	6
	MTQ ノイズ	1	53	30
	正常	21	42	379

まず Beat ノイズであるのに、正常と判断されたノイズスペクトルが図（4.14）である。後ろにある黒いノイズスペクトルは、正しく Beat ノイズと判断されたノイズスペクトルである。これを見ると Beat ノイズであるが、正常と判断されたスペクトルは、10 Hz 以上の部分でピークがあまりたっていないのも特徴である。これより、10 Hz 以上の情報が Beat ノイズの判断基準に入っていることが予想され、0 Hz から 10 Hz に絞って学習させることが、Beat ノイズの精度向上となると思われた。

次に MTQ ノイズであるのに正常と判断されたスペクトルが、図（4.15）である。後ろにある黒いスペクトルは MTQ ノイズと判断されたものである。これを見ると、381 Hz でのスペクトルが低いものが正常と判断されていることがわかる。よって 381 Hz に判断基準が向くような工夫をするべきだと考えられる。加えて、MTQ ノイズの中には、127Hz とその倍波でピークがたっているが、ピークが低いものも含まれていた。そのような MTQ ノイズが正常な判断を妨げていると考えられた。

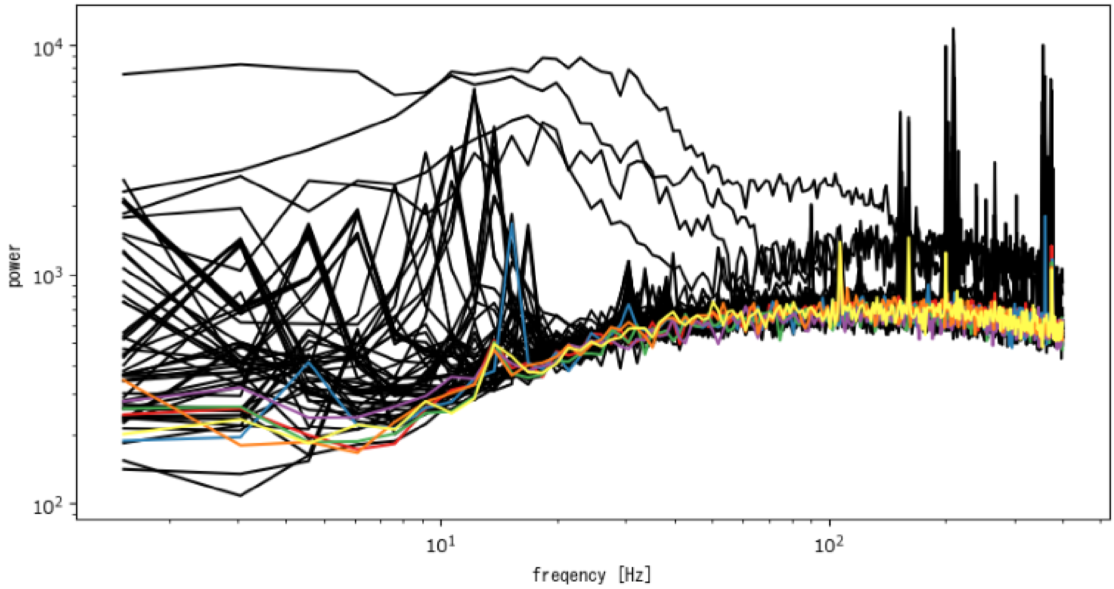

図 4.14. 正常と判断されたビートノイズのスペクトル

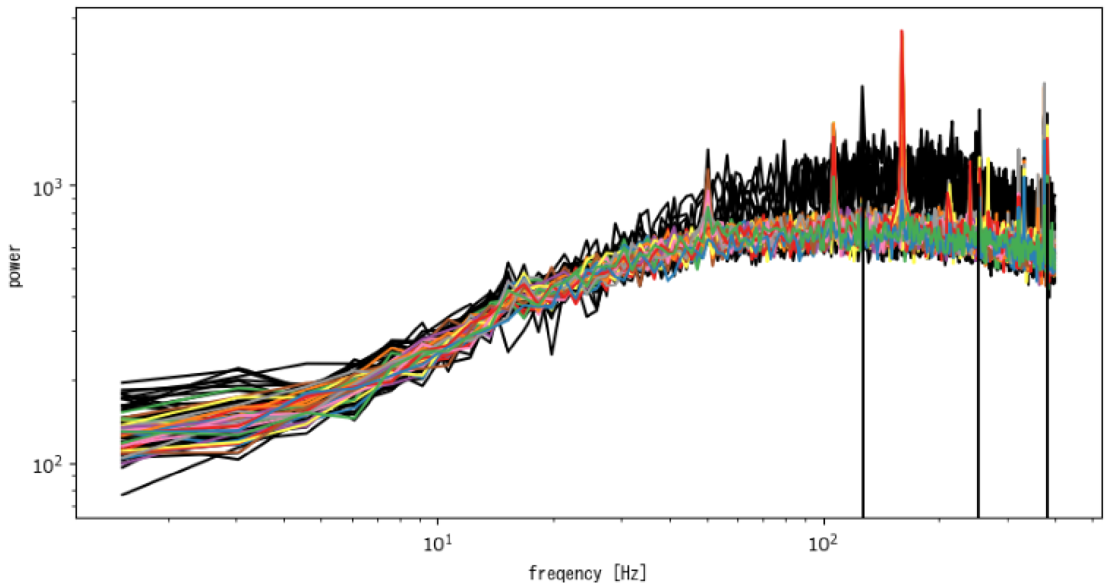

図 4.15. 正常と判断された MTQ ノイズのスペクトル

再構成誤差法を用いた MTQ ノイズと Beat ノイズの検出

次に再構成誤差を利用し、MTQ ノイズと Beat ノイズが検出できるかを検証した。これは、今回は異常のノイズスペクトルが、多種多様なスペクトルであることを考慮し、教師あり学習では検知できなかった異常スペクトルを、再構成誤差法で検知できる可能性を考え、検証した。再構成誤差法には PCA を用いた方法と Kmeans を用いた方法があるが、今回は再構成誤差法そのものが有効であるかを検証するため、一旦 Kmeans を用いた再構成誤差法を行った。このとき再構成誤差法でも、正常データセットを念のために平坦化する必要があると考え、以下の手順で、最適なクラスタの数 k と平均化するサンプル数 n を探索した。またデータセットは図（4.16）のように分けて検証した。以下ではそれぞれの手順について説明する。

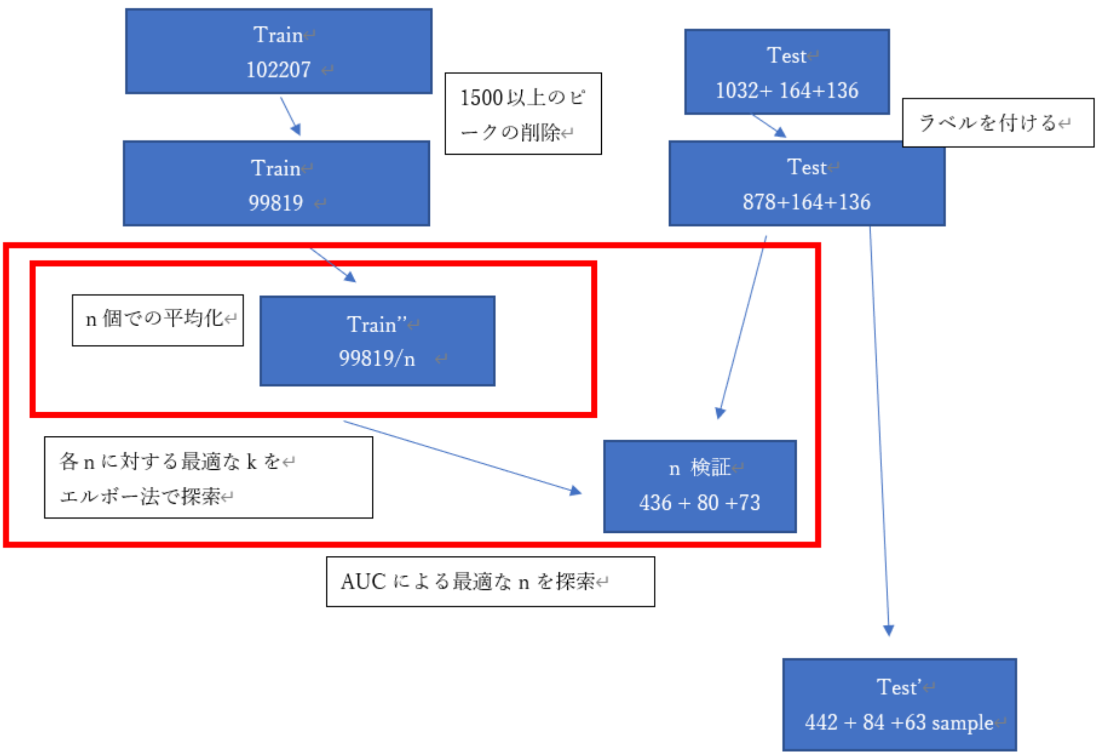

図 4.16. 再構成誤差法を用いた同時検知のサンプル

Step 1 n=1 個のデータを Train’ からランダムに抽出し、それの平均を Train” に追加した。これをすべてのサンプルが Train’ からなくなるまで、繰り返した.

Step 2 Train” データを kmens 法を用いて、k=1 個にクラスタリングした。このときのクラスタ中心を $C = \{c_1, c_2, \ldots, c_k\}$ とする。

Step 3 Train” データセットから 1 サンプルを取り出し、C の中から一番距離の近い点 c_i を選ぶ。そのサンプルと c_i の距離である再構成誤差を、そのサンプルの異常度とした。これをすべての Train” データセットのサンプルで繰り返し、すべての異常度の和を求める。

Step 4 Step2、step3 を k=1,2,3...6 で行う。それぞれの k に対する異常度の和をもちいて、エルボー法により、n に対する、最適な k=$\hat{k}$ を求めた。

Step 5 k=$\hat{k}$ にクラスタリングした時のクラスタ中心 $C = \{c_1, c_2, \ldots, c_k\}$ とし、step3 と同様に、n 検証データセットの各サンプルの異常度を求めた。その後異常度をもとに AUC 値を求めた。

Step 6 step2 から step5 を n=1 と n=10,20,...60 で行い、それぞれの n に対する AUC を求め、一番 AUC 値が高い n=$\hat{n}$ を求めた。

Step 7 $\hat{n}$ と $\hat{k}$ に対して、Test’ データセットに対して、異常度を step2 のように求める。このときの各サンプルの異常度と ROC 曲線を求めた。

今回は、Step1 で教師データである正常なデータを、設定した平均化するサンプル数 n について平坦化をし、Step2-Step6 ではｎ検証用データを、平坦化したデータに対するクラスタに振り分け、再構成誤差を求めた。またStep4 では、最適なクラスタ数 k を求めるため、エルボー法を用いた。最適なクラスタ数を求める方法は、いくつか提案されているが、今回は計算量を少なくするため一番シンプルなエルボー法を用いた。ただしクラスタリングする際に気を付けるのが、正規化の方法だ。通常正規化をする場合、教師データとテストデータそれぞれ正規化を行う。これは、教師データとテストデータが同じようなカテゴリ分布となるからだ。しかし今回は、教師データには正常データのみが入り、テストデータには異常データと正常データが含まれる。これをそれぞれ異なるデータスケーリングを行うと、二つのデータは別物となり、学習の精度が劣化する。ゆえにこのような場合で、データのスケーリングをする場合は、教師データでのスケーリングをテストデータに適応させる、もしくはスケーリングを行わない方が良いことが、実験的に分かった。異常より Step 4 で求めた k をもとに、Step6 では最適な平均化するサンプル数 n を求めた。その結果、最適なパラメータは k が 2,n が 2 となった。

Step7 では、その結果を用いて実際に異常検知をした。step7 でのテストデータに対する異常度をプロットした図が、図（4.17）である。そしてこのときの ROC 曲線が図（4.18）となる。図（4.17）を見ると、青い線で区切られているのが、データセットの区分である。一番左が正常データセット、真ん中が MTQ ノイズデータセット、右が Beat ノイズデータセットである。Beat ノイズは異常度が高いが、MTQ ノイズの異常度が低いことがわかる。これについて、再構成誤差は正常なスペクトルの平均スペクトルからの距離で定まる。このとき、正常なスペクトルはピークが低いので、ピークが多くたっている場合や、高いピークが立っている場合は、再構成誤差が高くなる。しかし、MTQ ノイズの場合は、127 Hz でピークが立っていたとしても、低いピークであることがあり、その場合は再構成誤差は低くなる。ゆえに再構成誤差は、MTQ ノイズを検出する方法として適していないことがわかる。これより MTQ ノイズは異常度が低く出力され、Beat ノイズは異常度が高く出力されてたので、ROC 曲線が途中で曲がったと考えられる。実際に、異常度の閾値を 3.0 としたとき、異常であるのに正常と判断されたスペクトルを見てみる。図（4.19）が正常と判断されたスペクトルであり、後ろにある黒いスペクトルは異常と判断されたスペクトルである。この図をみると、低周波や MTQ ノイズに限らず、全体的にパワーの値が高いものや、高いピークが立っているものが異常と判断されていることがわかる。以上より再構成誤差法は Beat ノイズに対してはある

程度有効であり、特に正常との差分が大きい 10 Hz 以下の周波数に対して再構成誤差法を実行することで、精度が上がることが予想される。

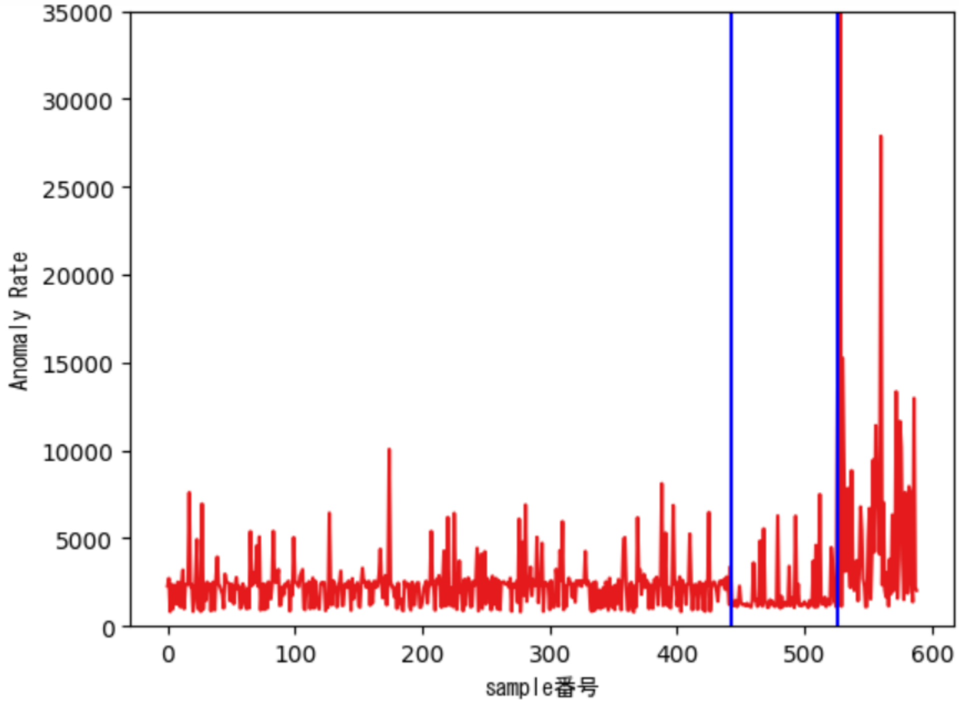

図 4.17. 同時検知における Kmeans の再構成誤差

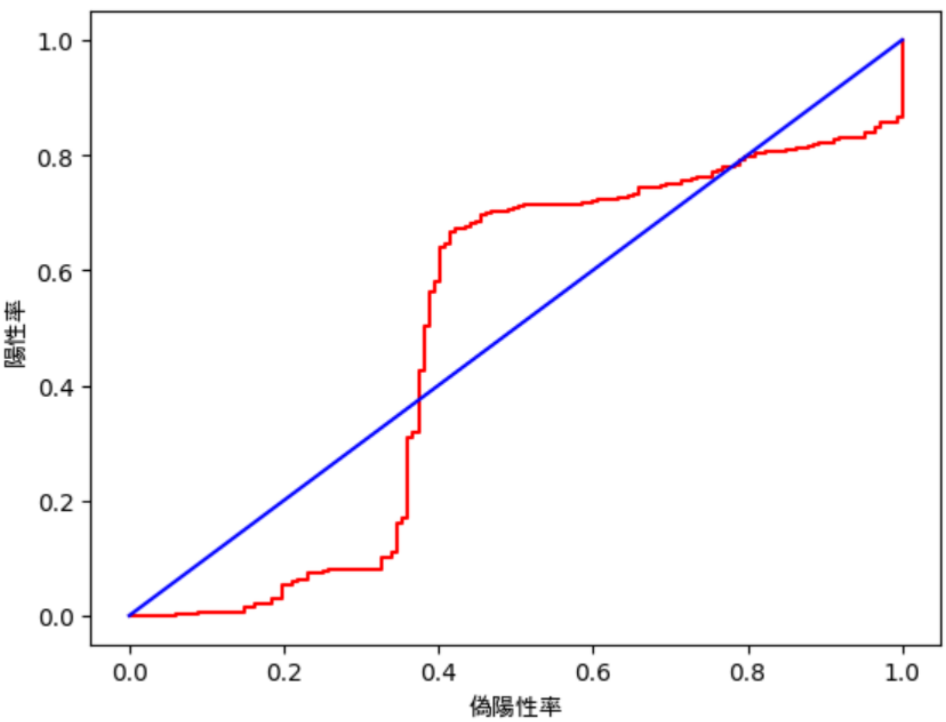

図 4.18. 同時検知における Kmeans の再構成誤差での ROC 曲線

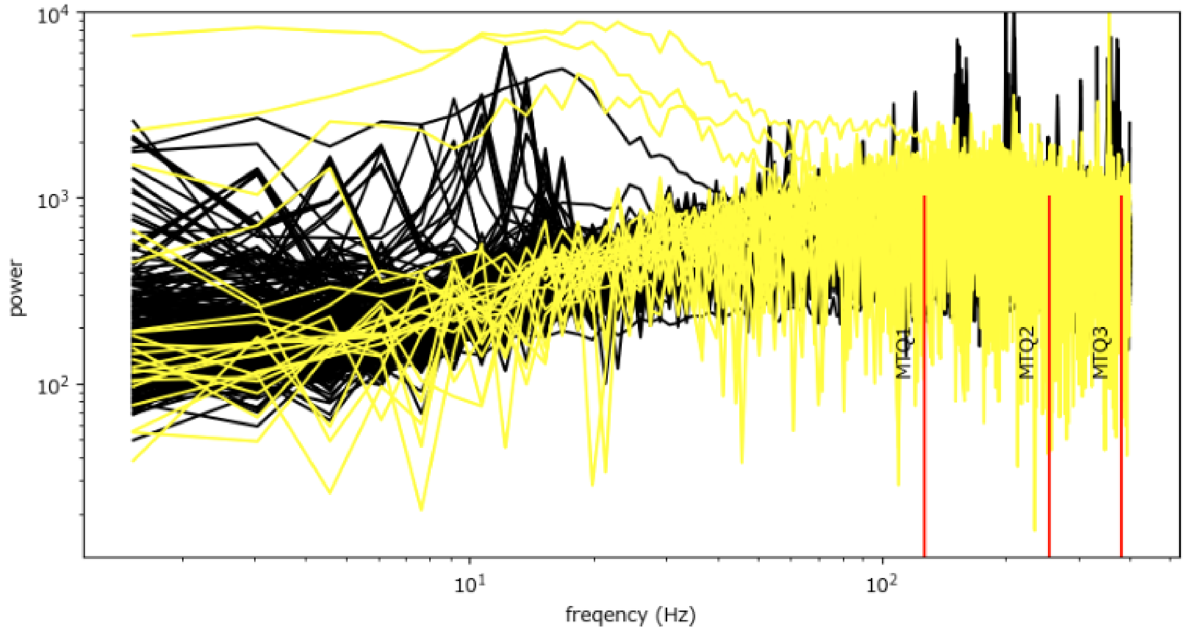

図 4.19. 同時検知における Kmeans の再構成誤差においての正常と判断された Beat ノイズ

以上より、再構成誤差法と教師あり学習での同時検出の結果を踏まえ、二つの異常ノイズスペクトルを別々に検出することで、より精度が良い検出器が作成されると考えた。

教師あり学習を用いた MTQ ノイズの検出

MTQ ノイズと Beat ノイズの同時検出の検証結果より、教師あり学習を用いた MTQ ノイズ検出器の作成を試みた。今回は 10 Hz から 400 Hz を学習、テストする周波数帯とした。これは Beat ノイズの有無が MTQ ノイズの検出に影響を及ぼさないような周波数帯域にすべきだからである。加えて、MTQ ノイズがうまく検出できない理由の一つに、MTQ ノイズには、ピークが低いノイズスペクトルも含まれていることがあった。これはピクセル 0、9、18、27 はピークが立ちやすいが、それ以外はピークが立ちにくいことが原因となる。MTQ ノイズは、MTQ が特定のシステム時に、すべてのピクセルで生じると予想されているので、ある時刻であるピクセルで MTQ ノイズが生じていることがわかればよい。よってピークが立ちやすい四つのピクセルに特化した、MTQ ノイズ検出器の作成を目標とした。

今回は図 (4.20) のようにサンプルを分け、モデルに対する最適なパラメータと平均化するサンプル数 n を探索した。この時 4.2.3 で行った手順と同様の手順を踏んだ。しかしこ今回、ピクセル数を 4 つのピクセルに限定したことで異常ノイズスペクトルのサンプル数が 100 サンプル程度となった。ゆえにパラメータ探索の精度や学習の精度が大幅に低下した。なので今回は、SMOTE をもちいて、異常サンプルを増やし、正常サンプル数とある程度均一な数にした。これらを踏まえ今回は、以下のような手順を取った。またモデルとして、SGDClassifier と LinearSVC を検討した。

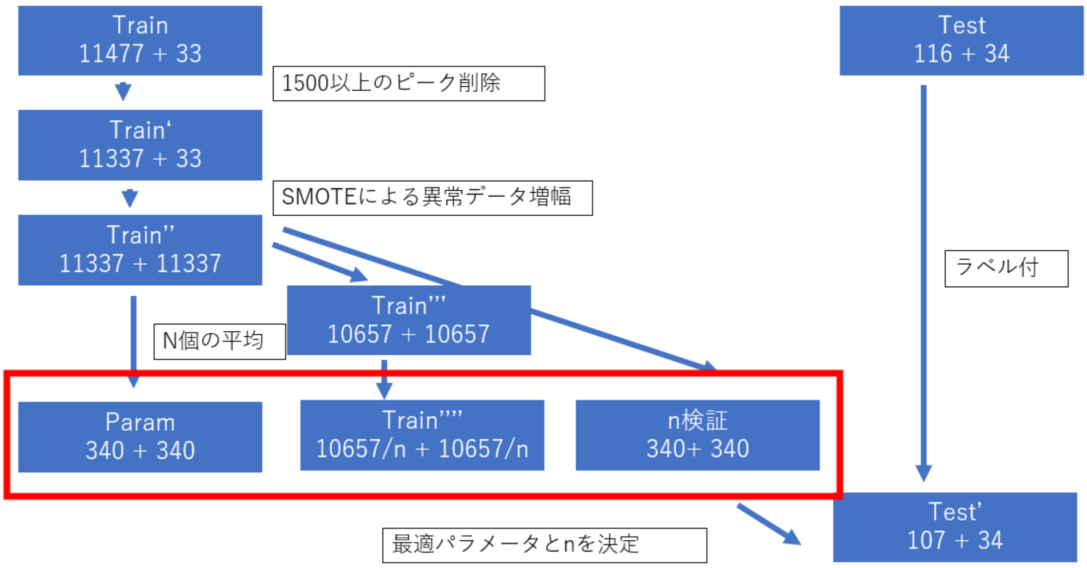

図 4.20. MTQ ノイズ検知におけるサンプル

Step 1　Train' のデータセットにある異常サンプルを SMOTE を用いて、作成・増幅した。

Step 2　Train データセットをパラメータ検証用データセット（Param）と Train' データセット n 検証データセットに分けた。

Step 3　n=1 個のデータを Train' からランダムに抽出し、それの平均を Train" に追加した。これをすべてのサンプルが Train' からなくなるまで、繰り返した。ただし残りの Test' データが n 未満になった段階で終了した。

Step 4　Train" のデータセットを SGD Classifier に学習させ、Param のデータセットを用いて、最良のパラメータを検証した。

Step 5　Step4 で得られたパラメータによる SGD Classifier に Train" のデータセットを学習させ、n 検証データセットを用い、F 値を求めた。

Step 6　n の値を n=10,20,...,200 と変化させながら、step2 から step4 を繰り返した。各 n に対し、Step5 で得られた F 値が一番高い $n=\hat{n}$ を求めた。

Step 7　step6 で得られた最適な $n=\hat{n}$ とその時の最適なパラメータを用いて、Test' のデータセットを学習、評価する。

今回の手順は §4.2 とほとんど同様である。しかし教師データが重度の不均一データとなっていたので、これを解消するため Step1 の段階で、Train データセット、Parameter データセット、n 検証用データセットを SMOTE でオーバーサンプリングしてから、分割した。

Step1 から Step4 でパラメータを検討した結果、LinearSVC ではパラメータの loss を hinge に、C を 1 にし、平均化するサンプル数を 1 にした。また SGDClassifier では、パラメータの penalty を l1、alpha を 1e-07 に、loss' を squared_hinge'、平均化するサンプル数を 1 にした。平均化するサンプル数が 1 となったことについて、今回ピークが立っているもののみが教師として与えられたので、平坦化する必要がなくなったためだと思われる。このときの Step6 で得られた各 n に対する F 値は、LinearSVM、SGDClassifier で、それぞれ図（4.21）と図（4.22）である。そして最終的な混合行列が表（4.5）と表（4.6）である。

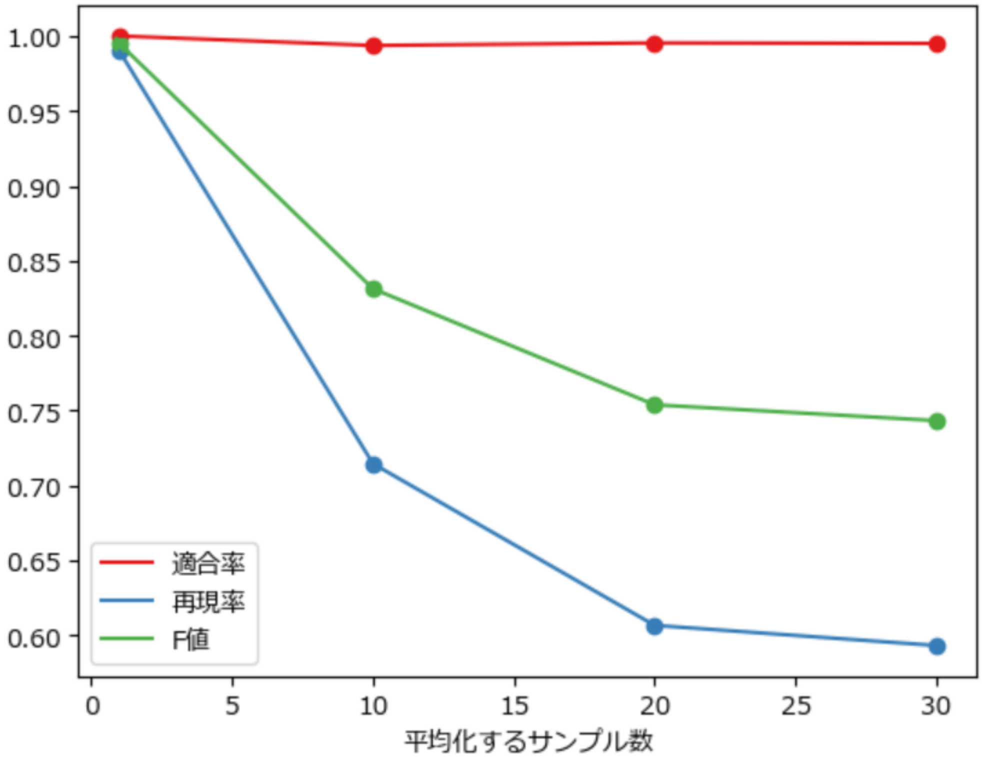

図 4.21. LinearSVC での平均化するサンプル数における F 値

表 4.5. LinearSVC での MTQ ノイズ検出の混合行列

		判断された結果	
		MTQ ノイズ	正常ノイズ
実際の結果	MTQ ノイズ	31	3
	正常ノイズ	8	99

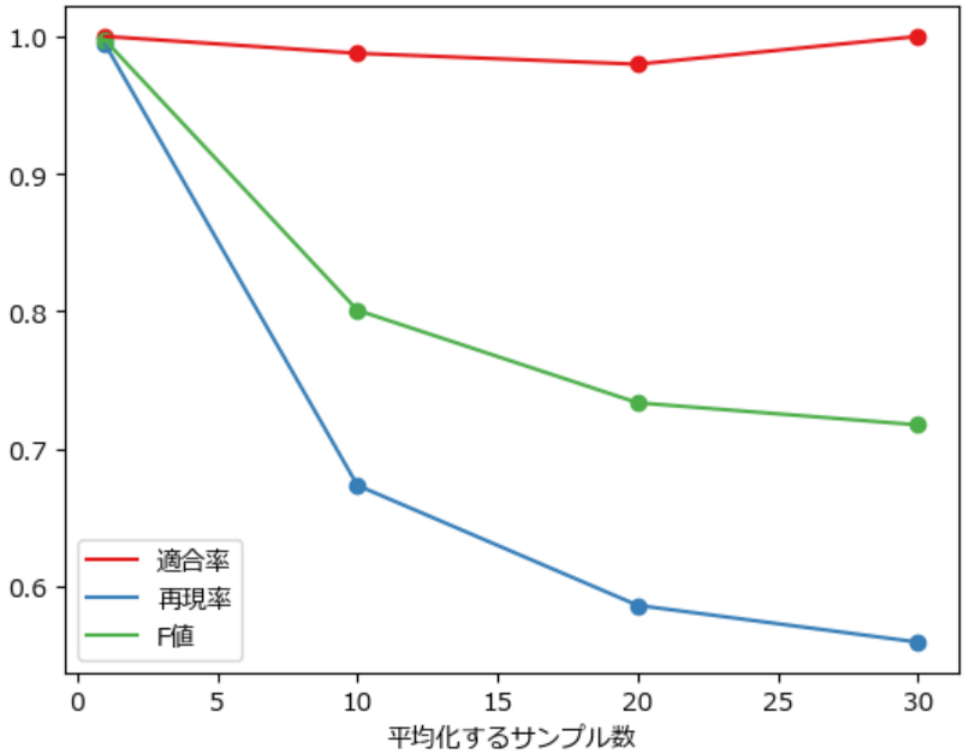

図 4.22. SGDClassifier での平均化するサンプル数における F 値

表 4.6. SGDClassifier での MTQ ノイズ検知の混合行列

		判断された結果	
		MTQ ノイズ	正常ノイズ
実際の結果	MTQ ノイズ	33	1
	正常ノイズ	13	94

これをみると、LinearSVC では陽性率は 0.91、SGDClassifier では陽性率は 0.97 となった。よって非常に良い精度での検出器が作成できた。加えて、SGDClassifier で分類できなかったスペクトルが図（4.23）である。これをみるとピークがほとんど立っていないスペクトルであった。これは MTQ が動作していても、ピークが低くなる条件下で作動してたと思われる。ゆえに、四つのピクセルに限定しても中にはピークが立っていないものもあるが、それは本来検出するべきではないので、ピークが立っているものはすべて検出できたといえる。

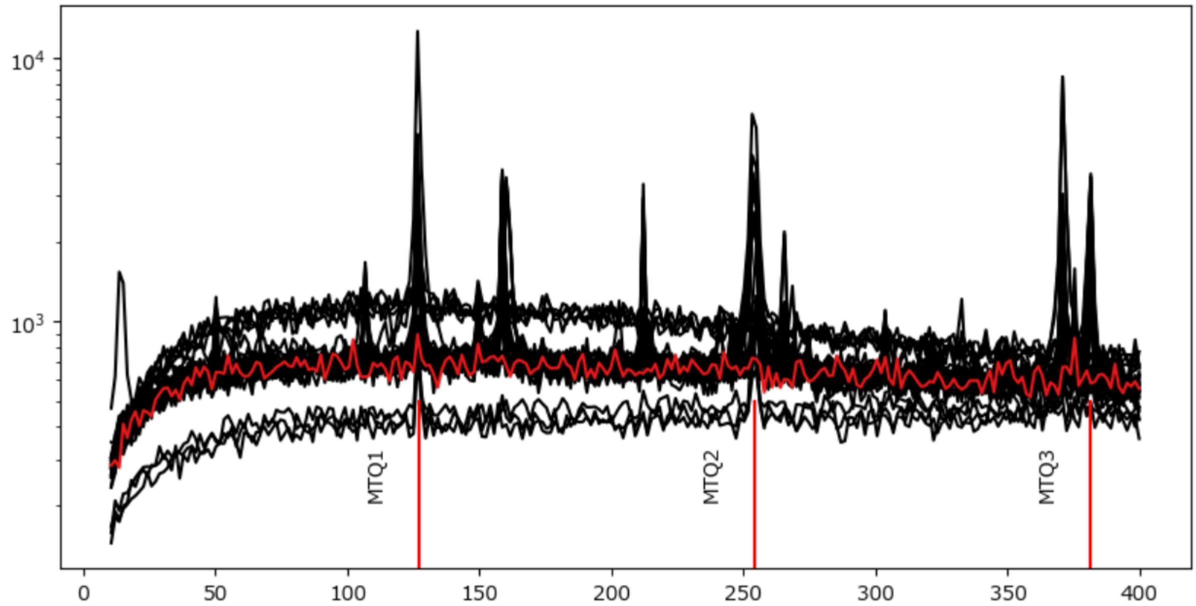

図 4.23. MTQ ノイズ検知で正常と判断された MTQ ノイズ

Beat ノイズの検出

最後に Beat ノイズの検出を行う検出器の作成を試みた。4.2.3 の結果より、Beat ノイズは教師あり学習、再構成誤差のどちらを利用して検出できることが分かった。しかし基本的に教師あり学習は、未知の異常データに対応しづらい。なぜなら未知の異常データがある場合、対応できない可能性があるからだ。このようなデメリットがあるので、今回は再構成誤差法を用いた。今回は再構成誤差の中でも PCA を用いた再構成誤差法でと、Kmeans を用いた再構成誤差法を検証した。

再構成誤差法を行った際の大まかな方針は、§4.2.3 と同様にした。ただし今回は、正常データのクラスタリングをしなかった。これは、スケーリングを行うと再構成誤差が小さくなるので、精度が劣化したからだ。加えて今回、9 次元はどれも同じようなデータスケールになっていたので、スケーリングの必要性がなかったからである。このとき、再構成誤差法を行った。まず今回用いたサンプルは図（4.24）のように分けた。そして図のような手順でのパラメータ探索の結果、PCA の場合は平均化するサンプル数は 2、次元数が 2 となった。そして Kmeans の場合は平均化するサンプル数は 1、次元数が 2 となった。この結果、PCA をもちいた再構成誤差が図（4.25）であり、Kmeans を用いた再構成誤差は図（4.26）となった。また ROC 曲線は図（4.27）であり、それぞれの AUC は表（4.7）となった。

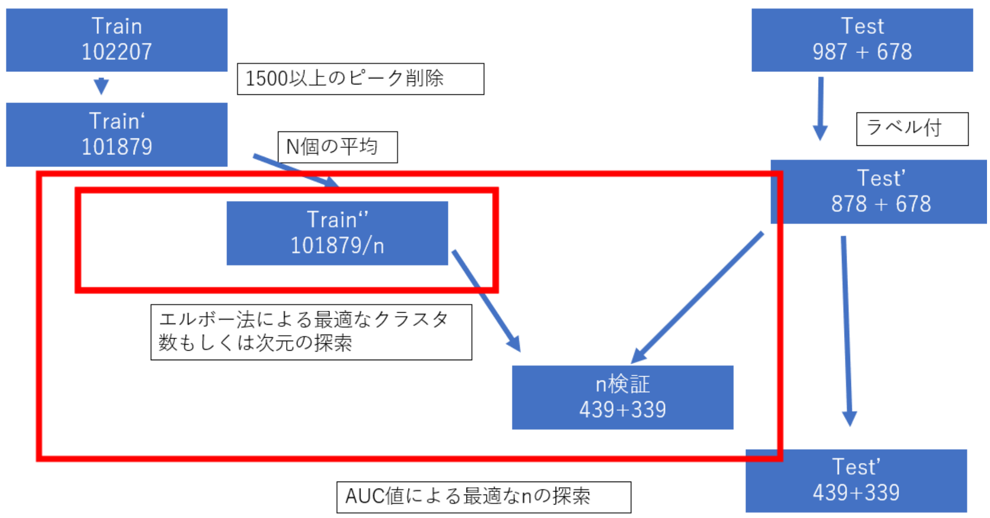

図 4.24. Beat ノイズ検知におけるサンプル

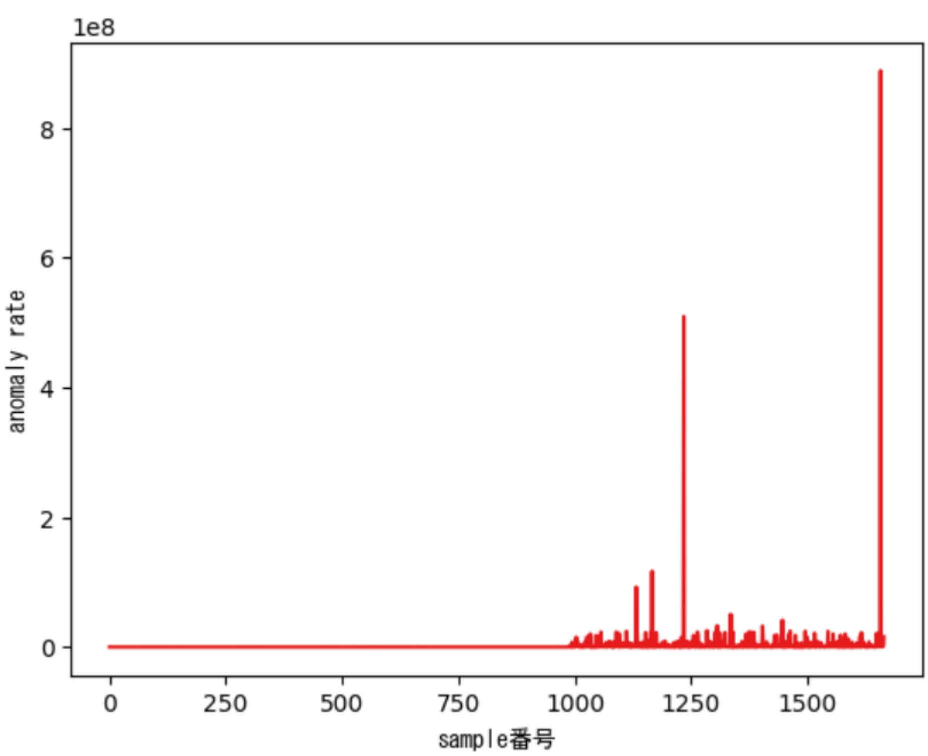

図 4.25. Beat ノイズ検知における PCA での再構成誤差

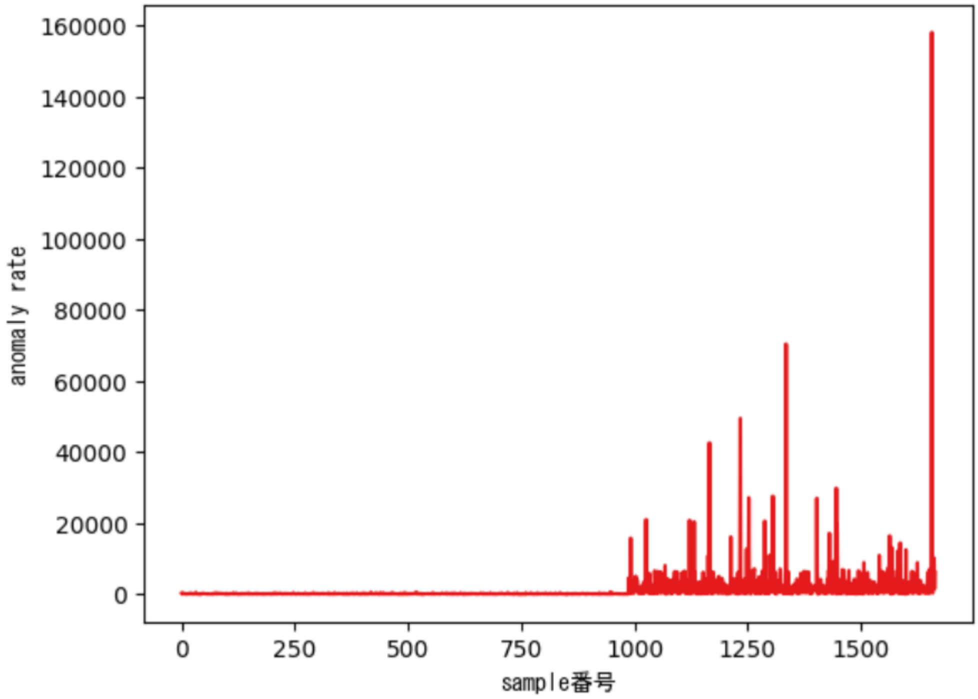

図 4.26. Beat ノイズ検知における Kmeans での再構成誤差

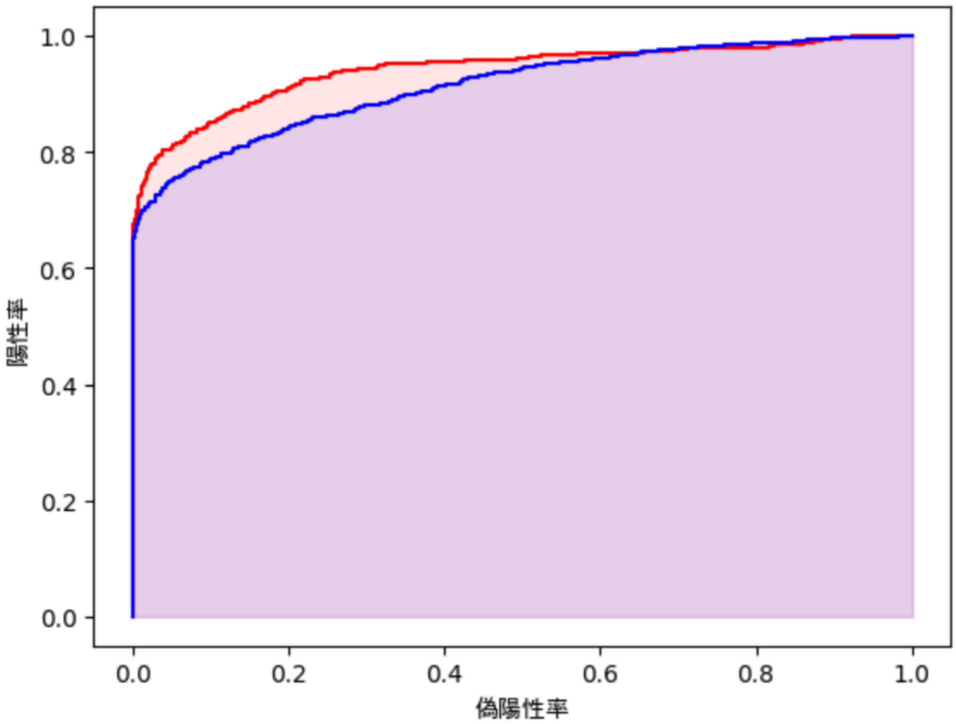

図 4.27. Beat ノイズ検知での ROC 曲線

表 4.7. Beat ノイズ検知に対する AUC 値

	Kmeans	PCA
AUR	0.942	0.914

これを見ると、Kmeans、PCA ともに再構成誤差はよい値を取った。

4.3　50 mK ステージの異常温度変化検知

4.3.1　現象の説明

まず *Resolve* 装置の検出器は、通常 50 mK に保たれている。しかし *Resolve* 装置に異常が発生したとき、もしくは ADR Recycle 等の動作が行われると、50 mK から温度が外れる。

ここで ADR とは 2.1.2 で述べたように、*Resolve* 装置に搭載されている冷却器の一種である。(23) そして ADR Recycle とは、ADR の断熱消磁サイクルを繰り返すために、再磁化をすることである。ADR が ADR Recycle をするとき、50 mK ステージは独特なパターンでの温度変化をする。図（4.28）は 2022 年 2 月 10 日 15:00 頃から生じた ADR Recycle である。このとき実際に、温度が急激に変化したのち、温度揺らぎを起こしながら減少していくパターンが見える。この独特の温度変化パターンは、他の ADR Recycle でも同様に生じていた。ゆえに、温度変化後の温度変化パターンを観測すれば、原因が ADR Recycle であるかを分類することができると予想される。

ゆえに今回は 50 mK ステージの温度を観測し、50 mK から外れた時刻を検出する。その後、50 mK から温度がずれた原因が正常な ADR Recycle か、それ以外の原因かを分類する分類器の作成を目標とする。加えて、このときの分類器は、失敗した ADR Recycle も異常と判断する必要がある。ADR Recycle が失敗すると、ADR Recycle による温度変化パターンが変化することが予想される。

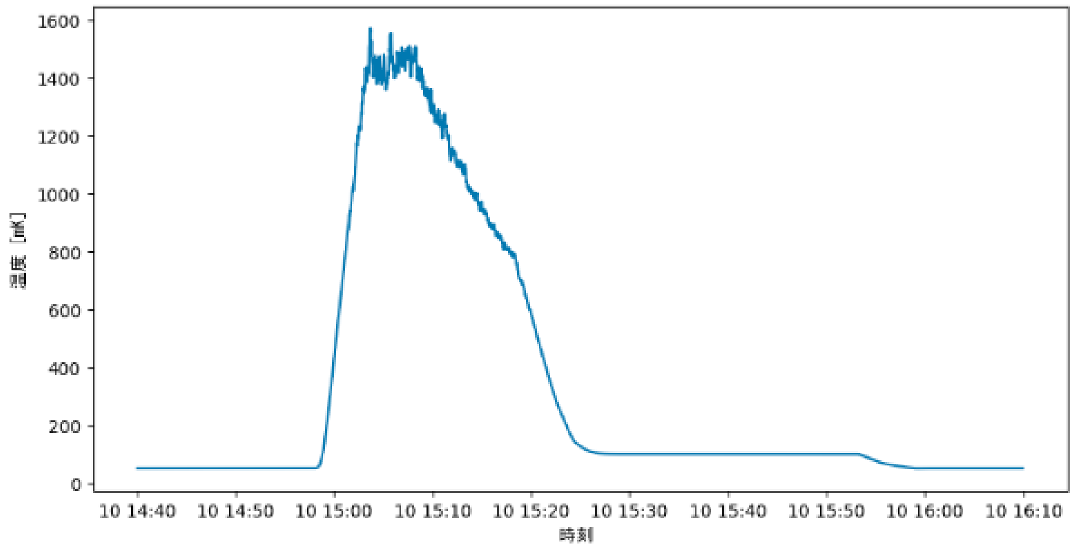

図 4.28. ADR Recycle1 発生時の例

4.3.2　要件分析

§4.3.1 を踏まえ、*Resolve* 装置の地上実験で得られた *Resolve* 装置の HK データに対し、統計的手法を用いて、50 mK から温度が変化した時刻を検知する。その後 50 mK からの温度変化のパターンを、機械学習的手法を用いて学習・分類することで、その温度変化の原因が正常な ADR Recycle か、失敗した ADR Recycle か、それ以外の原因不明な温度変化であるのかを特定する。

4.3.3　開発と性能

要件分析に伴い、今回は機械学習、統計的手法を用いた時系列解析を用いて、50 mK ステージから温度が外れた時刻を検知した。そして変化時刻からの温度変化パターンより、なにが変化の原因であるかを分類することを目標とした。

全体の流れとしては、最初に機械学習の精度を上げるため、データセットの前処理を行った。そして、前処理を行ったデータセットを用いて、以下の 3 つの手順に伴い検出器の開発を行った。

①. 50 mK ステージの温度変化点検知アルゴリズムの検討
②. 50 mK ステージの温度変化点検知
③. 50 mK ステージの温度変化パターン分類

ただし③の 50 mK ステージの温度変化パターン分類に関しては、今回は目標とする分類器の作成を達成ができなかったため、省略する。以下では各過程の具体的な作業、それぞれの過程の結果を述べる。

データセットの前処理

まず、今回用いたデータセットについて説明する。今回用いたデータセットは、*Resolve* 装置の地上実験で、2022/1/24 01:18:40 から 2022/06/17 02:36:13 までの期間で得られた、*Resolve* 装置の HK データのうち、50 mK ステージの温度データを用いた。50 mK ステージの温度データには、モニター用とコントロール用の二つの温度があるが、今回は実際の温度に沿った検知を行いたいので、モニター用の温度データをもとに、変化点検知を行った。この期間で得られた 50 mK ステージの温度データは、424 ファイルであった。それぞれびファイルには、およそ 12000 秒から 14000 秒程度の区間の時系列データが入っており、全体のサンプル数は、5786691 sample 存在した。

次にデータセットの前処理について説明する。今回の時系列データは各サンプル間の時間間隔が一定ではなかった。しかし時系列データを統計的手法で扱う際は、一定の時間間隔で並べられた時系列データを扱う必要がある。ゆえに今回は、1 秒ごとの間隔をもつ時系列データに変換した。具体的には、ある時刻の秒以下の値を切り捨てたとき、同じ時刻となるデータの平均を、その時刻での値とした。加えて、与えられた時系列データは連続する時系列データではなく、計測されていない区間も確認された。その計測されていない区間を補うため、計測されていない区間は 1 mK の値を持つように補完した。このとき 1 mK という実際の測定値は取らない値で補完することで、欠損値であることを明確にした。その結果、サンプル数は 12446251 sample となった。

50 mK ステージの温度変化点検知アルゴリズムの検討

データセットの前処理の後、50 mK ステージの温度変化が起きる時刻の検知を行った。今回温度変化時刻の検知にあたり、微分値による閾値を用いた検知と変化点検知アルゴリズムによる検知を検討した。それにあたり、ADR Recycle が起き、かつ ADR Recycle 後温度が変化している区画を選んで、検討した。

まず微分値による閾値を用いた検知は、時系列データを微分、すなわち一秒ごとの差分を取り、その値がある閾値を越したら検出するというものだ。これに従い、抽出した区間での微分値をプロットしたものが図（4.29）である。

これを見ると、確かに閾値が ADR Recycle 周辺で上がっているが ADR Recycle 中での微分地が検出したい変化点時刻での微分値よりも大きい。そのため、変化した時刻に対して、さらにルールを付け、検出する必要がある。ルールベースの検知を行う場合、ルールを破る未知の温度変化に対応できない可能性がある。加えて ADR Recycle 後の微妙な温度の変化は、見かけ上確認できない小さい値での微分値であり、変化を検知することが難しいと思われた。

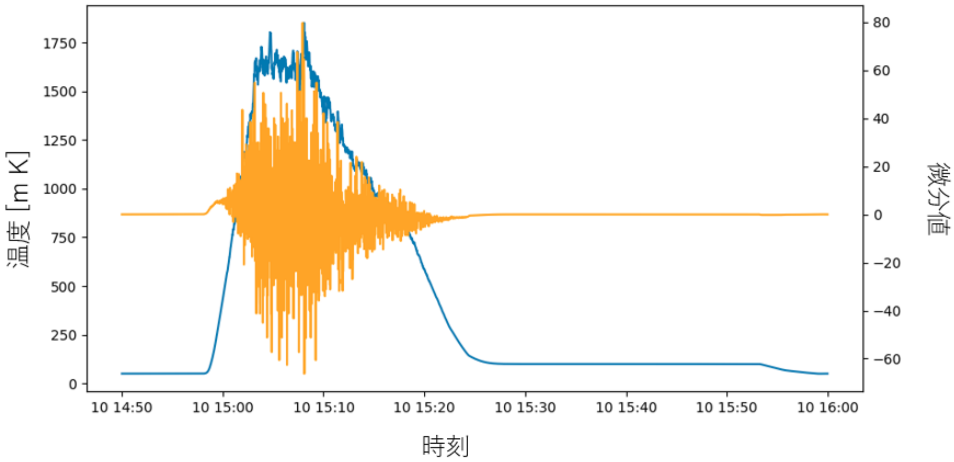

図 4.29. 微分値による温度変化時刻検知

次に変化点検知アルゴリズムによる検知を検討した。今回は変化点検知アルゴリズムの中でも自己回帰モデルによる異常検知を行った。それにあたり、*changefinder* モジュールを用いて、AR(1) での自己回帰モデルを用いた異常検知を行った。これを ADR Recycle 周辺で行った異常値をプロットした図が図（4.30）である。

これを見ると、ADR Recycle 開始時刻、ADR Recycle 後の変化時刻で、確かに変化点スコアが高く出力されていることがわかった。これより以降では、ChangeFineder を用いての変化点検知を行った。しかしこれを見ると、見かけ上変化していない部分に対しても変化点スコアが高く出力されていることがわかる。これは *changefinder* でのパラメータの取り方によるものであり、実際に検知する際はさらなる、細かいパラメータの調節が必要となることがうかがえた。

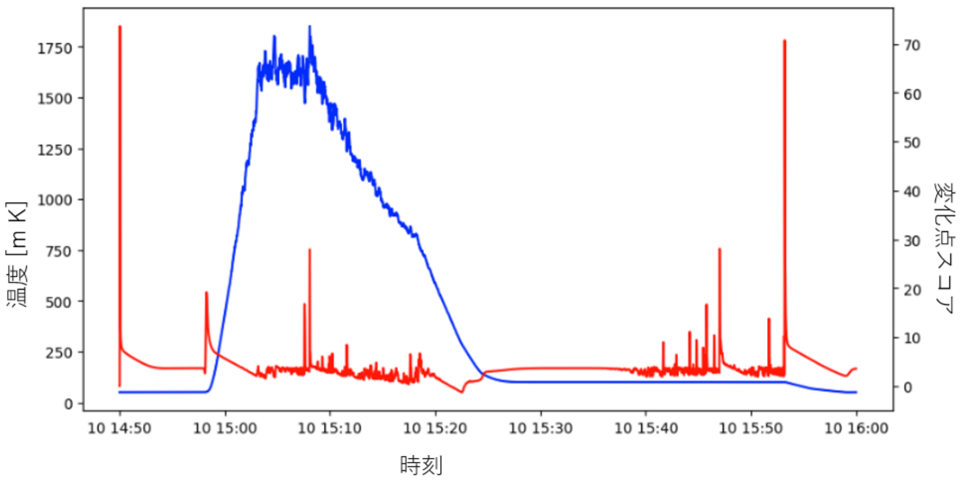

図 4.30. 変化点検知による温度変化時刻検知

50 mK ステージの温度変化時刻検知

§4.3.3 より、50 mK ステージの温度変化時刻を検知することができた。ゆえにこの節では、検討したアルゴリズムである *changefinder* を用いて、12446251 秒間での 50 mK の温度変化時刻の検知を行うことを目標とした。ChangeFineder を用いた変化点検知では以下のような手順を踏み、検知を行った。以下では各手順の概要と説明を

行う。ただし各 Step での区間の長さは、図（4.31）のようにした。

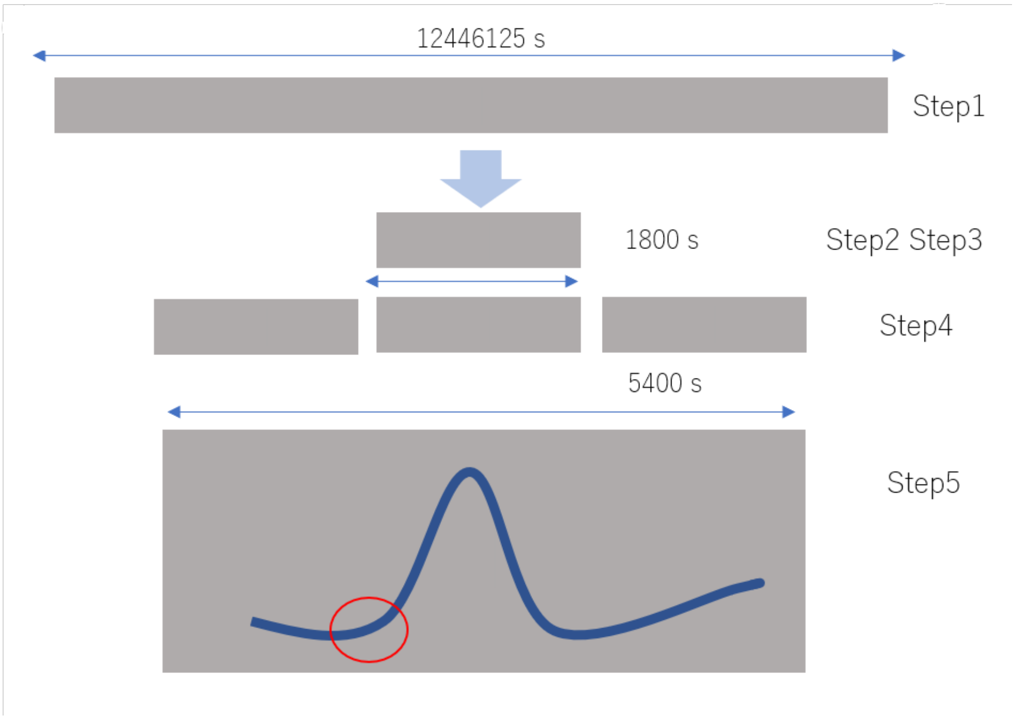

図 4.31. 各手順の区間の長さ

Step 1 12446251 秒間の時系列温度データを $1800s \times 6914$ ごとの時間に区切った。これを一区間とした。

Step 2 一区間の温度データのうち、温度が 52 mK 以上かつ微分値が 0 以下となる区間を抽出した。

Step 3 Step2 で抽出した区間のうち、常に 500 mK 以上となる区間を除いた。

Step 4 Step3 で抽出した区間の前後の区間をつなげた 7200 s を新たな区間とした。この区間に対して、Kmeans を用いてクラスタリングを行い、2 クラスに分類した。そのうちの 0 番クラスタをに属する区間を抽出した。

Step 5 Step4 で抽出した区間の時系列データに対しローパスフィルタをかけ、ChangeFineder を適応した。そして算出した変化点スコアに対し、ある一定の値を超えた時刻を抽出した。ただしこの時、温度データが 1 mK になる時刻の前後 10 s は、変化点スコアに 0 をかけた。これにより、補完したデータに対し、変化点検知を行わないようにした。

Step 6 Step5 で検出した時刻から、1000s 前から 100s 後までを再抽出した。そして再抽出した区間において、再度変化点検知を行った。そこで、Step5 で検出した時刻から前後 30s で、500 以上の変化点スコアを持つものを検出した。

まず Step 1 ではデータを 1800 s ごとに分けた。これは変化点検知をする際、長時間に対し、一括で変化点検知を行うと、精度が悪くなることが確認されたからである。また 1800 s という時間の区切りは、ADR Recycle が検知した前後区間の 7200 s 区間に収まるようにするためである。ADR Recycle による温度変化パターンの周期は、およそ 1800 s である。ゆえに、図（4.32）のように、区間ギリギリの部分で検知しても、ADR Recycle が入るようにした。しかしこのように区切り、6914 区間すべてに対し変化点検知を行うと、時間がかかる。ゆえに Step2 か

ら Step4 にかけて変化点検知を行う区間を絞った。

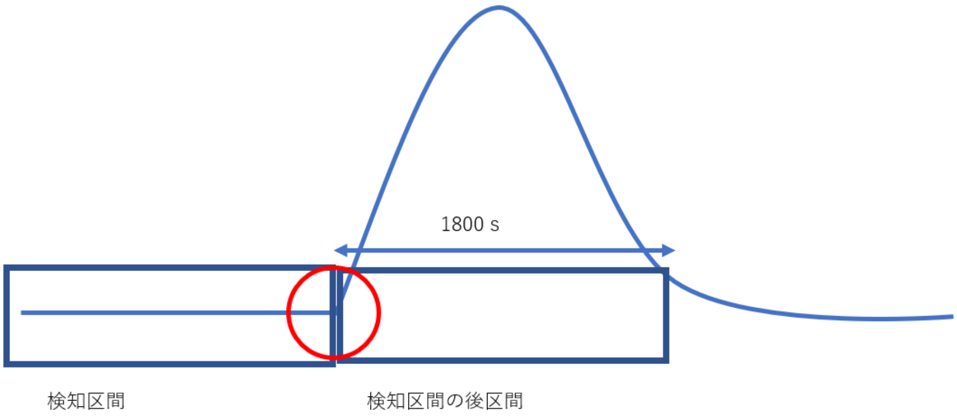

図 4.32. 1800 秒の区間の概略図

次に Step2、Step3 について説明する。Step2 Step3 では条件付けによる変化部位の絞りこみをした。Step2 では、温度が 52 mK 以上となる点を選んだ。50 mK ステージは温度が精密に保たれており、何か特別な操作をしない限り温度が 52 mK を超えることはない。ゆえに 52 mK 以上の点がある区間を選んだ。加えて Step2 では、微分値が 0 ではない点を抽出した。これは今回は変化点を見たいので、変化点検知をかける前に、温度変化をしていない区間を除くためである。ここまでの条件で摘出されたものを確認すると、中には 50 mK ステージが冷却途中で、500 mK よりも高い温度で一定となっている区間があった。その一例が図（4.33）である。この状態では ADR Recycle が確認されなかったため、これを取り除くことにした。よって、Step3 では補足したデータ以外に、500 mK 以下のデータがないデータ区間を取り除いた。

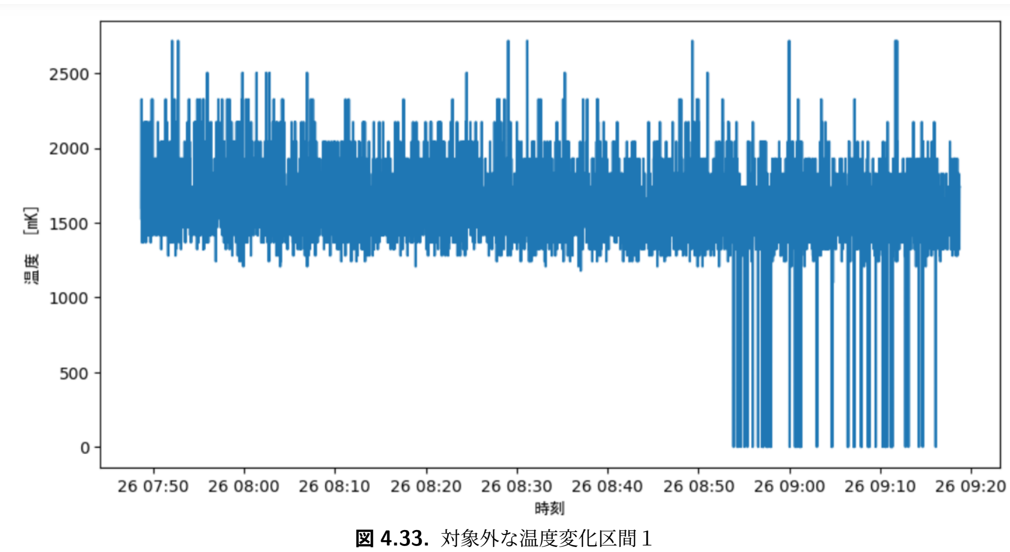

図 4.33. 対象外な温度変化区間 1

Step4 では、Step2、Step3 では明らかにほかの温度変化と異なるものを除いた。具体的には Step3 の時点で図（4.34）のような区間が多く残っていた。この区間は、データの欠損値が多い区間であるか、冷却中の 50 mK ステージの温度であると考えられた。ゆえに、このような区間に対して変化点検知を行うことは今後の分類の精度低下につながるので、正規化をしたのち、Kmeans を用いたクラスタリングを行い、除去を試みた。その結果ある冷却中

の区間とそれ以外の区間をほとんど正確にクラスタリングされた。

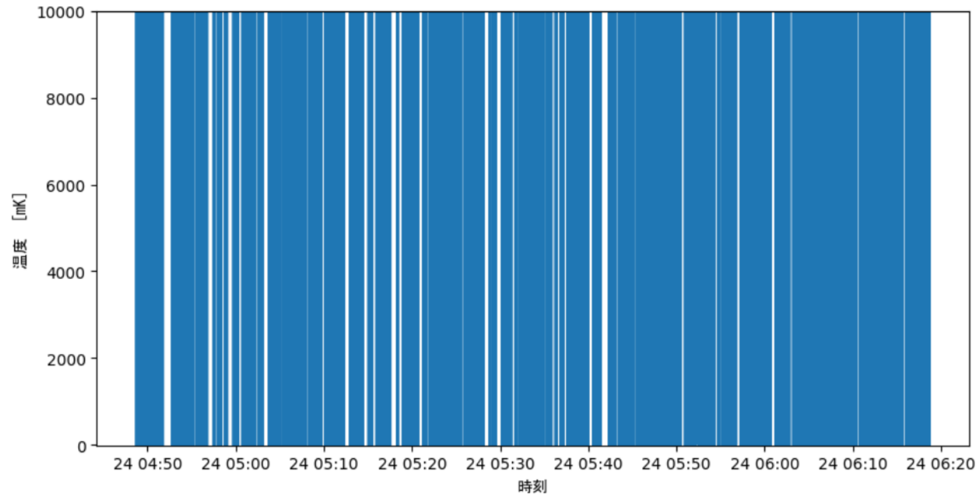

図 4.34. 対象外な温度変化区間 2

Step5 以降では、Step4 までで抽出した区間に対して、変化点検知をかけた。ただし Step5 では、補完したデータに変化点検知が反応しないように、変化点スコアに対し、補完したデータの前後 30 s では 0 をかけた。これにより、ある程度の補完したデータによる温度変化検知が除けた。

Step5 では、抽出した区間の温度データに対し、直接 ChangeFineder を行うのではなく、温度データに対しローパスフィルタをかけてから、ChangeFineder を行った。これは図（4.35）のような ADR Recycle が生じるとき、ADR Recycle の開始前の温度に、ノイズが多く含まれている。この場合微小な変化が多く、ADR Recycle 開始時の微小な変化が ChangeFineder では検知されなかった。ゆえに、変化点検知を正確に行うためには、ノイズを取り除く必要があり、そのためにローパスフィルタをかけた。

ローパスフィルタとは、あるデータをフーリエ変換したときに算出されるスペクトルに対し、ある一定以上の周波数に対するパワーをなくし、逆フーリエ変換をすることである。これにより、高周波のデータが除去されるので、今回の温度ノイズのような細かい変化をする温度データをなだらかにすることが出来る。ローパスフィルタをするにあたり、閾値を 0.05 Hz としてそれ以上の周波数を除いた。その時のフィルタをかけた図が図（4.36）における黄色い線である。そしてローパスフィルタをかけた温度変化に対し、変化点検知を行ったのが図（4.37）である。これより ADR Recycle 時に変化点スコアが高く出力されていることがわかる。

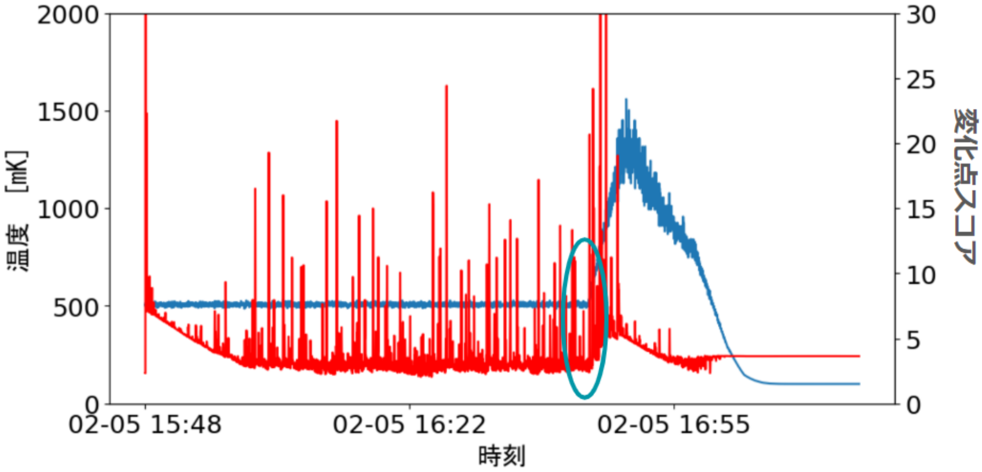

図 4.35. ローパスフィルタ前の変化点スコア

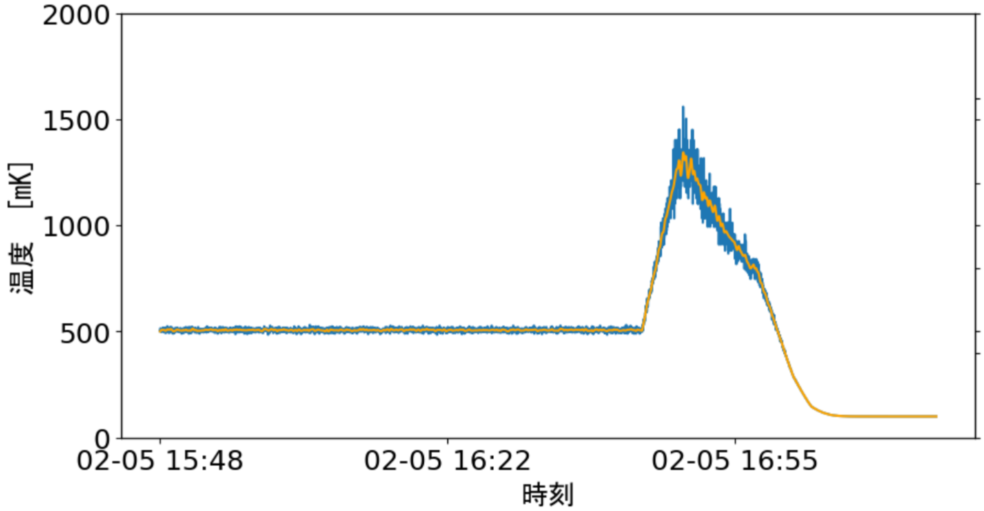

図 4.36. ローパスフィル後の温度変化グラフ

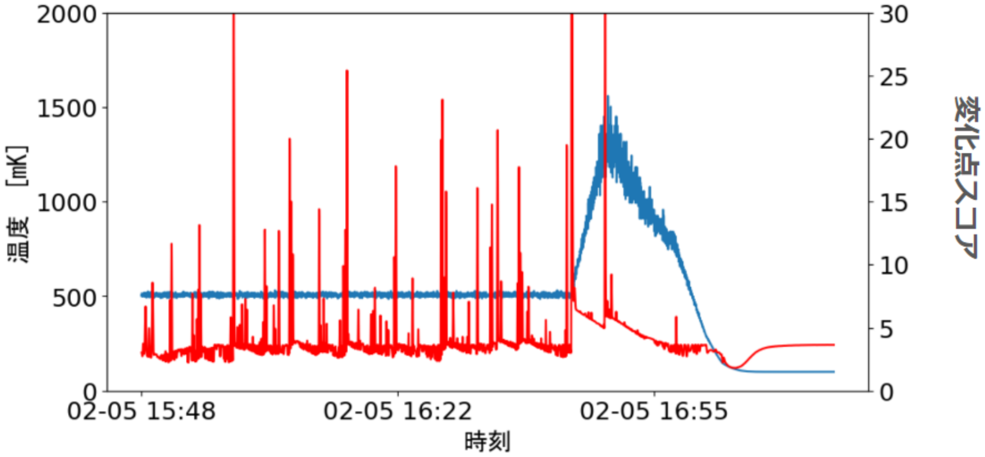

図 4.37. ローパスフィル後の温度変化グラフへの変化点検知

Step5 で、検知された区間は 143 区間で、変化点は 660 点となった。ただし、*changefinder* のパラメータは r を 0.05、order を 1、smooth を 3 とした。このとき r は SDAR モデルの忘却パラメータであり、この値が小さい、非定常性を持つようになる。加えて、比較的大きな変化のみを検知するになった。そして smooth は 3.3.2 での SDAR アルゴリズムにおける $2L$ の値をあらわし、今回は短い区間での変化点検知を行いたかったので、最小値にした。ただ、図 4.37 をみると、温度揺らぎのところで、変化点検知が過剰に反応していることがわかる。これらの変化点検知は本来避けたい。ゆえに温度揺らぎによる変化点検知を避けるため、Step6 を行った。

この温度揺らぎに対する過剰変化点検知、検知した部分を取り除くため、Step6 では、検知した時刻に対し前 1000s、後ろ 100s を取り、合計 1100s の区間で再度変化点検知を行った。この時は *changefinder* のパラメータを、r を 0.01、smooth を 15 とした。これにより温度揺らぎでの部分は定常性を持つので、検知点スコアが図（4.39）のように相対的に小さくなる傾向が見れた。これにより閾値を 50 に設定することで、温度揺らぎによる変化点検知の数が削減できた。

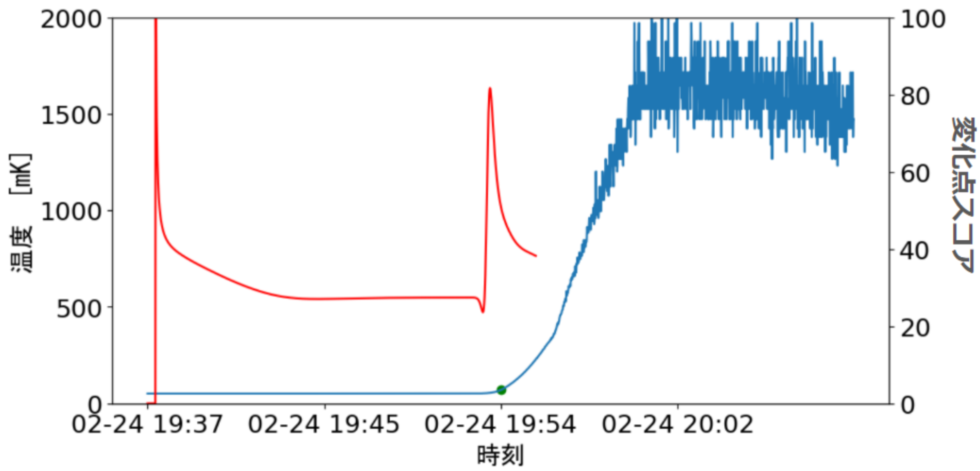

図 4.38. Step6 での ADR Recycle 前の変化点検知

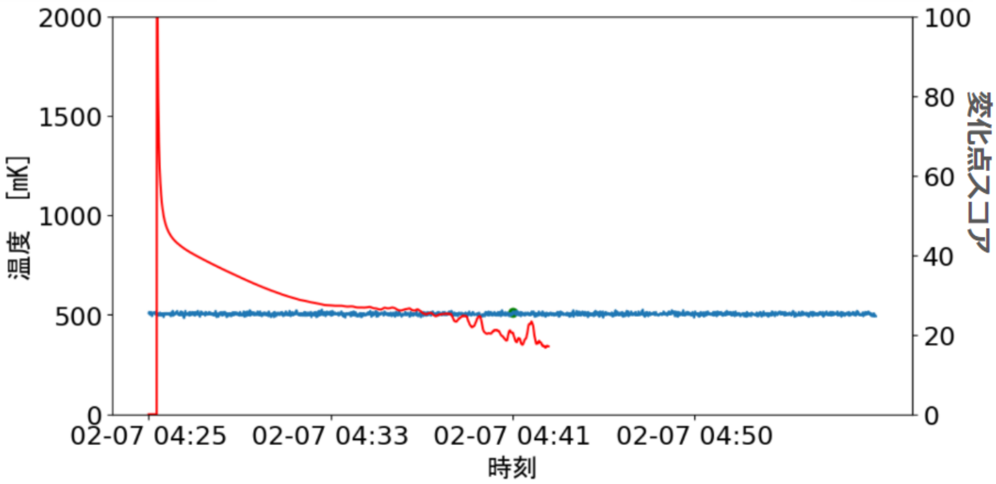

図 4.39. Step6 での異常変化点前の変化点検 2

最終的に、Step 6 までで得られた変化点時刻は 289 点となった。この中にはやはり、正常な ADR Recycle による温度変化と ADR Recycle 以外の温度変化が入っていた。また ADR Recycle 以外の温度変化の中には、原因が不明の異常な温度変化と、Step6 で取り除けなかった温度揺らぎによる温度変化、Step5 で取り除けなかった補完したデータによる温度変化、そしてグリッチによる温度変化が入っていた。ここでグリッチによる温度変化とは、50 mK ステージは侵入したエネルギーに敏感であり、わずかでも宇宙線などのエネルギーが入ると温度が急激に上昇する。例えば図（4.40）がグリッチによる温度変化の一例である。割合としては、100 のサンプルをランダムに選んだ時、異常な温度変化が 9、グリッチによる温度変化が 22、ADR Recycle による温度変化が 16、温度揺らぎによる温度変化が 23、補完したデータによる温度変化が 30 となった。これを見ると、補完しているデータによる温度変化検知が多くなっているので、補完したデータ付近に 0 をかける範囲を長くするなどの、対応が必要になる。また、今後の目標は、検出器を改良し、温度揺らぎによる温度変化検知を取り除くこと、そして取り除いた変化点検知から、正常な ADR Recycle による温度変化、失敗した ADR Recycle による温度変化、異常な温度変化、グリッチによる温度変化を分類する分類器の作成が必要となる。

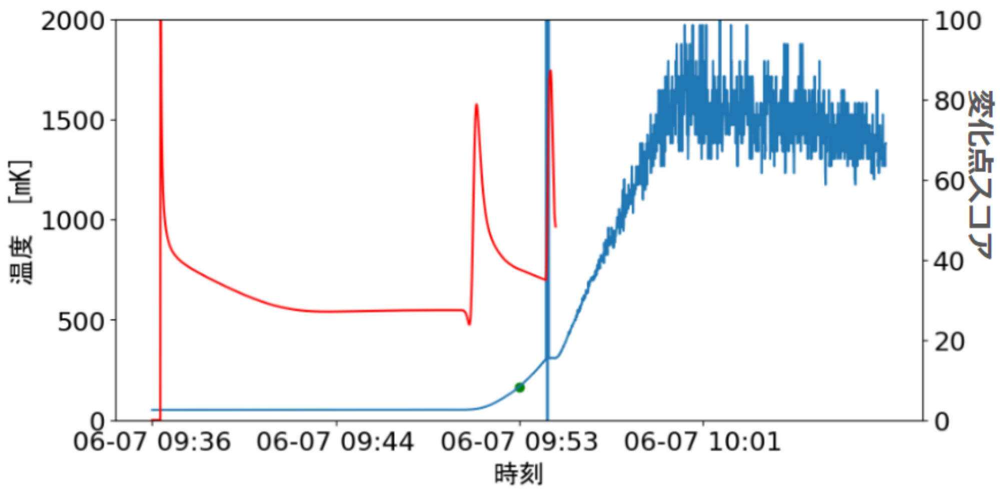

図 4.40. グリッチによる温度変化の例

第 5 章

結論

Contents

5.1　本研究の到達点

本論文では *Resolve* 装置に関する異常に対し、異常検知アルゴリズムを開発することを大きな目標とした。対象となる異常は、従来の閾値を用いた検知では、異常検知が困難なものであり、本論文ではノイズスペクトルの異常検知、50 mK ステージの異常温度変化検知という、二つの異常検知を行った。本研究の目的は、異常検知アルゴリズムを開発するにあたり、機械学習的手法による、異常検知が有効であるかを検証することであった。よって §4 での結果を踏まえ、ATMOS を補完するアルゴリズムとして機械学習的手法が有効であったかの、まとめを以下で述べる。

まずノイズスペクトルの異常検知での目標は、*Resolve* 装置のノイズスペクトルから、異常なノイズスペクトルとなる時刻を検知することであった。この異常は、従来の閾値を用いた異常検知では検出できなかった異常であった。そこで今回は機械学習的手法を用いて、対象となる二つのノイズ、MTQ ノイズと Beat ノイズに対して検出を試みた。MTQ ノイズには、教師あり学習である SGDClassifier を中心とした異常検知アルゴリズムを検討した。この際、異常サンプルが少ないため、SMOTE というライブラリを用いて、異常サンプルをオーバーサンプリングしてから、学習させせた。その結果、陽性率 0.97 での異常検知を可能とした。Beat ノイズには、半教師あり学習を用いた再構成誤差法を中心とした、異常検知アルゴリズムを検討した。半教師学習は、教師あり学習に比べ、未知の異常に対応しやすい異常アルゴリズムであるというメリットがある。これを用いて異常検知アルゴリズムを検討した結果、AUC 値 0.94 での検知となった。

次に 50 mK の異常検知での目標は、精密に 50 mK に保たれる必要がある検出器の温度に対し、異常な原因で温度が変化する時刻を検知することであった。このアルゴリズムの開発にあたり、本論文では 50 mK ステージが温度変化をする時刻の検出アルゴリズム開発までを行った。今回は自己回帰モデルを用いた変化点検知法を中心として、温度変化時刻の検出を試みた。その結果、ADR Recycle による温度変化時刻、2 mK 程度の緩やかな温度変化時刻の両方に対して反応する、温度変化時刻検知アルゴリズムの開発に成功した。またこの際、温度揺らぎに対しては、二段階での異常検知アルゴリズムを取ることで、過剰に温度揺らぎに対して反応する時刻を取り除けた。しかしそれでも、除けれない区間が残ってしまったので、改善は必要である。

以上を踏まえ、本論文ではノイズスペクトルの異常検知システムに対しては、機械学習的手法が有用であることが示せた。50 mK ステージに関しては、機械学習的手法を応用する場面まで至らなかった。しかしその前段階とな

る、温度変化検知アルゴリズムの開発の道筋はつけられた。

5.2　今後の課題

今後の課題としては、主に 2 つあげられる。一つ目が、50 mK 系での温度異常検知アルゴリズムの改良である。二つ目が、異常検知アルゴリズムのユーザーへのシステム開発である。

まず 50 mK 系での温度異常検知アルゴリズムでは、正確に温度変化時刻の検出がされた。しかし、現在の変化点検知では、補完したデータに対する変化点検知や、温度揺らぎによる温度変化検知など、本来検知したくない温度揺らぎによる温度変化も検知してしまっている。ゆえにこの検知を取り除くアルゴリズムを付け加えるが必要となる。加えて、今回検知した温度変化時刻に対し、失敗した ADR Recycle や、原因不明の温度変化時刻を検知することが最終的な目標である。これらを変化時刻からの温度変化パターンから、分類する検出器の開発も行う必要がある。

次に異常検知後のユーザーへ伝達するシステムの開発である。まず本研究大きな目標である、ATMOS を補完する異常検知システムの開発するにあたり、設計した計画は以下のとおりである。

①. 異常検知アルゴリズムの開発
②. 異常検知からユーザーへのシステム開発
③. GPU CPU を用いた異常検知の高速化

②では、①で開発した異常検知アルゴリズムに対し、異常データを検知した後にユーザに伝えるまでのシステム開発を目標としている。そして③では、②までのシステムに対し、より高速に運用できるように、CPU・GPU を用いたシステムの高速化を目標としている。ゆえにこれを XRISM 衛星で運用するためには、本研究で検討したアルゴリズムを用いて異常検知をした後、その異常はユーザーへ伝えるシステムの開発が次の目標である。

付録 A

補遺

Contents

この章では、本文で省略した数式や、前提として用いた理論などを説明する。

A.1 ノイズレコードの種類一覧

以下では本論文で使用した 8k noise spec 以外のノイズスペクトルについて述べる。

Noise record

Noise record は、波形ノイズデータである。コマンドの有無によらず、CPU に余裕がある限り常時、1024 サンプルの長さからなる Noise record を、1 セットとして記録する。Noise record は一つの CPU に対し、1 ピクセルのデータを一秒に一回取得する。X線マイクロカロリメータは 36 ピクセルに対し、4 つの CPU で呼び出すため、各ピクセルのノイズレコードは 9 秒に 1 回取得される。Noise record は、任意の時間帯で 1k noise spec が作成できる。しかしデータ量が少ないので、短時間の変動は追えない。

1k noise spec

1k noise spec は、周波数帯域が 12.2 Hz から 6.25 kHz で取得した、ノイズスペクトルデータである。取得方法は、コマンドで命令した時から、1024 サンプルの長さからなるノイズレコードを 1 セットとして記録する。サンプリングのセット数は 100 セットであり、サンプリングし終えた後、100 セットそれぞれをフーリエ変換し、平均してノイズスペクトルを作成してタンプする。

Sample dump

Sample dump は波形ノイズデータと波形ノイズデータを微分したデータ、さらに Anti-co 検出器のデータを一セットとしたノイズデータである。50 セット、もしくは 25 セットをサンプリングのセット数とし、これを一つの sample dump としている。コマンドで命令した時から、X線パルスの有無に関係なく取得される。全 pixel が同時に取得されるので、時間空間で相関するノイズの診断に有効である。またこれをフーリエ変換することで、12.2Hz

- 6.25kHz のノイズスペクトルも得る。

Wave-form ring buffler dump

Wave-form ring buffler dump(WFRB dump) は波形データと波形データの微分データ、そして Anti-co 検出器のデータを取得する方法である。10 セットを WFRB dump のセット数としている。取得方法は、コマンドで命令した時から、各 PSP カードごとに 2 ピクセル分を収集記録する。一回にとるサンプリングの長さは、場合ごとに異なる。

表 A.1. ノイズの種類一覧

ノイズの名前	含まれるデータ	周波数帯域	長さ	サンプルのセット数
noise record	波形データ 波形の微分データ	12.2 Hz	1024	任意
1k noise spec	スペクトル	12.2 Hz - 6.25 kHz	1024	100
8k nosie spec	スペクトル	1.5 Hz - 6.25 kHz	8192	50 or 10
sanple ump	波形データ 波形の微分データ	12.2 Hz - 6.25 kHz	1024	50 or 25

A.2　σ と半値全幅の関係

まつ半値全幅とは、正規分布の最大値の半分の値を取るところの幅となる。これを式で表すと

$$G_{\mu,\sigma}\left(\mu+\frac{FWHM}{2}\right)=\frac{1}{2}G_{\mu,\sigma} \tag{A.2.0.1}$$

となる。ただしこのとき $G_{\mu,\sigma}$ は正規分布を表し、

$$G_{\mu,\sigma}=\frac{1}{\sigma\sqrt{2\pi}}\exp-\frac{(x-\mu)^2}{2\sigma^2} \tag{A.2.0.2}$$

とする。これより、式 A.2.0.1 に式 A.2.0.2 を代入すると、

$$FWHM=2\sigma\sqrt{2\log 2}\quad\simeq 2.35\sigma \tag{A.2.0.3}$$

となる。

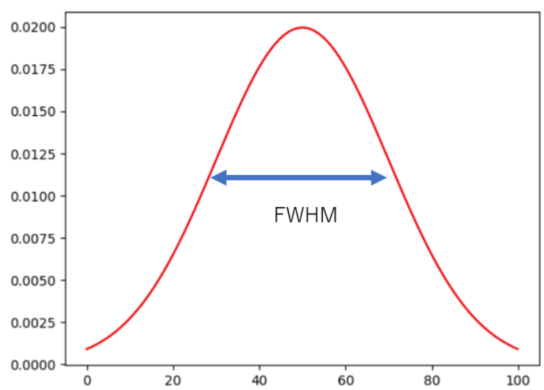

図 A.1. FWMP の例

A.3　損失関数

損失関数とは、制限付き最適化問題を、無制約最適化問題に変換して解くために用いる関数である。この方法では、制限に対し満たさないとペナルティが生じる、ペナルティ関数を目的関数に加える。これにより制限付きの最適化問題を、無制約最適化問題に変換する。機械学習での教師あり学習を行う際は、損失関数を用いて最適化問題を解くことが多い。

損失関数は、分類損失と回帰損失の二つが存在する。以下ではデータ (x, y) が与えられた際の、予想値 $f(x)$ に対する損失関数を考える。

まず回帰損失関数の中で代表的なものが、L1 損失、L2 損失である。L1 損失とは、二つのデータの絶対値誤差を考える。つまりこのときの損失関数は

$$\mathrm{loss} = |f(x) - y| \tag{A.3.0.1}$$

と定義される。次に L2 損失とは、二つのデータの二乗誤差を考え、

$$\mathrm{loss} = \frac{1}{2}(f(x) - y)^2 \tag{A.3.0.2}$$

と定義される。これはデータの差分を二乗しているので、離れれば離れるほど差が大きくなる。ゆえに L1 損失よりも影響が大きくなる。

次に分類損失を説明する。代表的なものにヒンジ損失、ログ損失などがある。まずヒンジ損失は、SVM などに一般的に用いられる損失関数である。ヒンジ関数は

$$L(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) \tag{A.3.0.3}$$

のように定義される。次にロジスティック損失は

$$L(y_i, f(x_i)) = \frac{1}{2}(y_i - f(x_i))^2 \tag{A.3.0.4}$$

のように定義される。これら二つの関数は、成功し境界面から大きく離れているときは、ペナルティを課さない。しかし判別に成功していても境界面付近ならば、小さなペナルティを課す。そして判別に失敗した場合は、さらに大きなペナルティを課す。

A.4　正則化

機械学習は一般的に損失関数を最小化するように、学習をする。しかし学習データに細かくフィッティングした複雑なモデルは、過学習を引き起こしやすい。ゆえに損失関数に、モデルの複雑さを表す項を付け加える必要がある。この関数を最小化にするように学習することで、シンプルなモデル選択が可能となる。これが正則化である。

正則化の関数として代表的なものが L1 正則化と L2 正則化である。まず L1 正則化とは

$$\alpha|\omega| \tag{A.4.0.1}$$

の項を、損失関数に付け加えることである。次に L2 正則化とは

$$\alpha|\omega|^2 \tag{A.4.0.2}$$

の項を、損失関数に付け加えることである。この二つの正則化の違いを（A.2）を見ながら説明する。この時、二つのパラメータ $\omega1$,$\omega2$ を決める場合を考える。真ん中にある高等線が、損失関数の当てはまりの良さを表し、遠ざかるほどに損失関数のスコアが悪くなる。

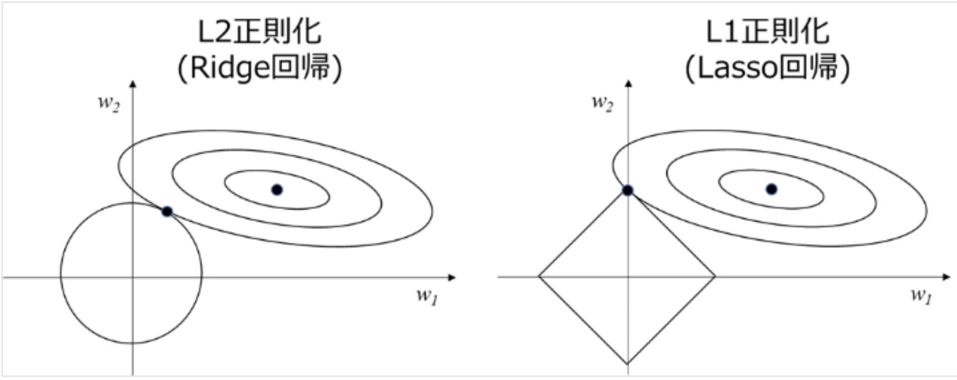

図 A.2. 正則化の概要図 (9)

まずＬ２正則化は、円状の等高線となる。つまりこの二つの円が接する、接点が最適なパラメータとなる。そして正則化の強さを表す α を大きくすると、正則化項の高等線は内側に移動する。また図をみると、$\omega 2$ 方向のほうが、$\omega 1$ 方向よりも損失関数の等高線の密度が高い。ゆえにこの場合 $\omega 2$ のほうが、寄与率が高い成分となるが、L2 正則化であれば、寄与率の大小関係を変えずに、係数選択をすることができる。

一方 L1 正則化は、ひし形の等高線となる。このとき、L2 正則化と同様に損失関数の等高線と接する点が、最適なパラメータとなる。そして多くの場合、ひし形の頂点が、解となる。これは、どちらか一つのパラメータが 0 になるので、L2 正則化よりも強い、過学習防止が期待される。

A.5　標本共分散行列

今 p 個の変数について、n 個の個体のデータを取り、これらを $n \times p$ 行列としてまとめたものを X とする。

$$X = (x_{i,j})_{1 \le i \le n, 1 \le p} = \begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \dots & \dots & \dots \\ x_{p,1} & \dots & x_{p,p} \end{pmatrix} \tag{A.5.0.1}$$

また、行列 X の行と列の和を

$$x_{.j} = \sum_{i=1}^{n} x_{i,j} (j = 1, \dots, p) \tag{A.5.0.2}$$

$$x_{i.} = \sum_{j=1}^{p} x_{i,j} (j = 1, \dots, n) \tag{A.5.0.3}$$

と表す。このとき変数 j の分散 $s_{j,j}$、変数 j、変数 i の共分散 $s_{j,k} (j \neq k)$ はそれぞれ

$$s_{j,j} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{i,j} - \bar{x_{.j}})^2 \tag{A.5.0.4}$$

$$s_{j,k} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{i,j} - \bar{x_{.j}})(x_{i,j} - \bar{x_{k.}}) \tag{A.5.0.5}$$

と表される。ただし $\bar{x_{.j}}$ は変数 j に対する n 個の平均である。これより、変数 j、変数 k の標本相関係数 $r_{j,k}$ は

$$r_{j,k} = \frac{s_{j,k}}{\sqrt{S_{j,j} S_{k,k}}} = \frac{\sum_{i=1}^{n} (x_{i,j} - \bar{x_{.j}})(x_{i,j} - \bar{x_{k.}})}{\sqrt{\sum_{i=1}^{n} (x_{i,j} - \bar{x_{.j}})^2 \sum_{i=1}^{n} (x_{i,j} - \bar{x_{k.}})^2}} \tag{A.5.0.6}$$

となる。これより標本分散共分散行列 S は

$$S = \begin{pmatrix} s_{1,1} & \dots & s_{1,p} \\ \dots & \dots & \dots \\ s_{p,1} & \dots & s_{p,p} \end{pmatrix} \tag{A.5.0.7}$$

と表される。

A.6　矩形波とフーリエ変換

まず矩形波とは図（A.3）のように周期的に、二つの値を行き来する波のことを指す。

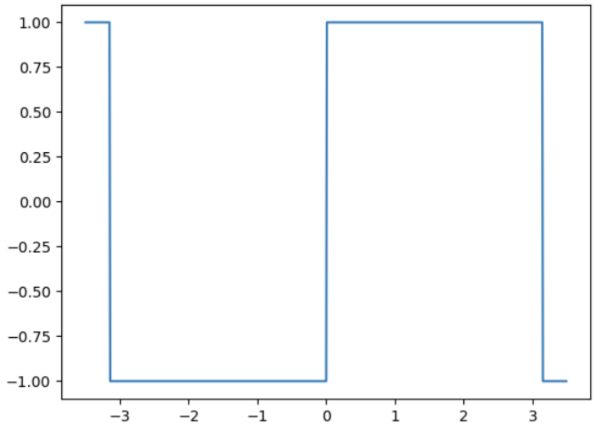

図 A.3. 矩形波の例

このとき矩形波をフーリエ変換する。まず任意の矩形波を

$$f(x) = \begin{cases} 1 + \pi > x > 0 \\ 0 - \pi < x < 0 \end{cases} \tag{A.6.0.1}$$

とする。このとき、フーリエ係数の式にあてはめると

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx = \frac{1}{\pi}(\int_{0}^{\pi} 1dx + \int_{-\pi}^{0} 0dx) = 1 \tag{A.6.0.2}$$

となる。また

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nx dx = \frac{1}{\pi}(\int_{0}^{\pi} \cos(nx)dx + \int_{-\pi}^{0} 0 \times \cos(nx)dx) = 0 \tag{A.6.0.3}$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nx dx = \frac{1}{\pi}(\int_{0}^{\pi} \sin(nx)dx + \int_{-\pi}^{0} 0 \times \sin(nx)dx) = \frac{1}{\pi}[\frac{-\cos nx}{n}]_{-\pi}^{0} = 0 or - \frac{2}{\pi n} \tag{A.6.0.4}$$

となる。これを用いると

$$f(x) = \frac{1}{2} + \sum_{n=1}^{\infty} \frac{-1 + (-1)^n}{n\pi \sin(nx)} = \frac{1}{2} - \frac{2}{\pi n}(\sin(x) + \frac{1}{3} sin(3x) + \dots) \tag{A.6.0.5}$$

と表される。つまり矩形波をフーリエ変換すると、元の周期の周波数の波に加え、その二倍波、三倍波が生じる。

謝辞

本論文を執筆するにあたり、多くの方にお世話になりました。中村研究室の皆様、並びに海老沢研究室の皆様をはじめとする JAXA 関係者の皆様に深く感謝します。

まず指導教官である中村正吾准教授には、研究室に配属された 3 月から、研究室内での定例ミーティングや誤差論のゼミ、学部生での宇宙論ゼミや学部生の進捗報告など様々なミーティングにお時間を取っていただき、指導していただきました。また研究以外でも大学院試験の際は志望動機書などの添削も取り組んでいただきました。私は文章を書くのが得意ではなく、拙い間違いを起こしやすいのですが、その点に関しても指摘していただき、先生に指摘されることでより客観的に自分の欠点を見つめなおすことができました。また勉強の際に生じた疑問点などにはいつも丁寧にお答えしていただき、参考資料なども探していただきました。中村先生のご指導はいつも丁寧であり、学部時代に表面的に学んだ内容も、中村先生のお話を聞くとより深みが増し、物理学の奥深さを学びました。来年度からは、他大学の研究室に移りますが、中村先生に教えていただいた物理学への姿勢をいつも大切にし、研究に取り組みたいと思っております。

次に JAXA で研究するにあたり、研究室で受け入れてくださった海老沢研教授にもお世話になりました。海老沢先生は来年度からの私の指導教官であり、学部四年のうちから技術習得生として卒業研究のための技術指導をしていただきました。海老沢先生には、JAXA での受け入れのための書類の手続きなどをしていただきました。加えて宇宙研で開講された X 線天文演習では、学部生で天文の知識がない私に、基礎的な部分から丁寧に天文学の指導していただきました。来年度から正式に海老沢研究室に所属となりますので、よろしくお願いいたします。

次に宇宙科学研究所の辻本匡弘准教授には本論文を執筆するにあたり、一番にお世話になりました。XRISM 衛星の異常検知というテーマは、私のなかで行いたかったデータサイエンスを活用するテーマかつ衛星開発にかかわるテーマであり、まさに自分がやりたいテーマそのもので、このような素晴らしいテーマを提案していただいたことに感謝しています。またお忙しい中、毎週水曜日に一時間個人ミーティングを取っていただき、自分の進捗を常に把握していただき、助言をしていただきました。機械学習は辻本先生の専門ではないのにもかかわらず、上手くデータが処理できない場合などは常に適切なアドバイスをしていただいたり、論文や参考書を勧めていただきました。さらに修士以降の研究のため、LiteBIRD の f2f meeting に参加することを勧めていただいたり、宇宙情報科学解析シンポジウムの参加なども勧めていただきました。ミーティングへの参加によって自分に足りないものが明確に分かったので、来年度から精進したいと思います。辻本先生とのミーティングでは、計画を立てることの大切さや自分の獲得目標を明確にすることの大切さなどを学びました。来年度からもよろしくお願いいたします。

次に宇宙科学研究所の中平さんには、研究データの可視化のために用いた InflaxDB や Grafana の使い方や、スクリプトの書き方を教えていただきました。また、InflaxDB や Grafana で上手くいかないとき、Slack 上で相談させていただいた時も、丁寧に教えていただきました。ありがとうございました。

次に愛媛大学の今村さんには低周波ノイズの異常の際に、お忙しい中 zoom でマイクロカロリメータのノイズなどについて教えていただきました。またその際に異常であるノイズデータをまとめていただきました。そのおかげで、研究がスムーズに進みました。ありがとうございました。

次に海老沢研究室、中村研究室の皆様にも大変お世話になりました。まずは中村研究室の皆様には普段のミー

ティングの際などに進捗報告している際にも、研究に対していつもコメントや助言をしていただきました。KEK の見学の案内や研究室の使い方、また初期のころから雑談などもしていただき、楽しい研究生活を送れました。10 月以降は金曜日のみ登校することも多かったですが、週に一度中村研究室の皆さんと食事をするのが楽しく、研究の疲れがとれました。来年度からは別の研究室に所属しますが、これからも仲良くしていただけると幸いです。また海老沢研究室の方々にも大変お世話になりました。海老沢研究室の皆様には、研究環境のセットアップのお手伝いや、研究で生じた疑問などを質問もよくさせていただきました。時には私の研究の調子なども部屋に聞きに来ていただき、本当に助かりました。ありがとうございました。それ以外にも、普段から雑談などもしていただき、楽しい研究生活を送れました。来年度から正式に所属することになりますが、どうぞよろしくお願いいたします。

参考文献

[1] 宇宙開発利用部会 X 線天文衛星「ひとみ」の異常事象に関する小委員会. X 線天文衛星 astro-h「ひとみ」 異常事象調査報告書. 国立研究開発法人　宇宙航空開発研究機構, 2016.

[2] XRISM 衛星公式サイト (2023/2/10) アクセス. `https://xrism.isas.jaxa.jp/about//`.

[3] NASA/GSFC ISAS/JAXA, ARD/JAXA. Astro-h 実験計画書第二分冊. 国立研究開発法人　宇宙航空開発研究機構, 2016.

[4] 武田佐和子. 修士論文. 埼玉大学, 2013.

[5] ExtruDesign(2022/10/15) アクセス. `https://extrudesign.com/stirling-cycle/`.

[6] 剛井手. 異常検知と変化検知 = Anomaly detection and change detection. 講談社, 2015.

[7] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. Vol. 109, No. 5, pp. 756–795.

[8] 矢入健久, 乾稔, 河原吉伸, 高田昇. 次元削減とクラスタリングによる宇宙機テレメトリ監視法. Vol. 59, No. 691, pp. 197–205.

[9] 機械学習で「分からん！」となりがちな正則化の図を分かりやすく解説 (2023/1/15) アクセス. `https://qiita.com/c60evaporator/items/784f0640004be4eefc51`.

[10] 乾稔, 矢入健久, 河原吉伸, 町田和雄. 次元削減の再構成誤差を用いた異常検知手法の比較. Vol. JSAI2009, pp. 1B12–1B12.

[11] Y. Ishisaki and others. Resolve instrument on x-ray astronomy recovery mission (XARM). Vol. 193, No. 5, pp. 991–995. Publisher: Springer US.

[12] 神田英蔵. 低温工学. コロナ社, 2000.

[13] 姫野俊一. 演習大学院入試問題. サイエンス社, 1992.

[14] 公益社団法人 日本冷凍空調学会 (2022/10/15) アクセス. `https://www.jsrae.or.jp/`.

[15] 日本統計学会. 統計学実践ワークブック. 学術図書出版社, 2020.

[16] 直希島田. 時系列解析：自己回帰型モデル・状態空間モデル・異常検知. Time series analysis. 共立出版, 2019.

[17] Andreas C. Müller. Python ではじめる機械学習：scikit-learn で学ぶ特徴量エンジニアリングと機械学習の基礎. Introduction to machine learning with Python : a guide for data scientists. オライリー・ジャパン, 2017.

[18] 塚本邦尊, 山田典一, 大澤文孝, 中山浩太郎, 松尾豊. 東京大学のデータサイエンティスト育成講座：Python で手を動かして学ぶデータ分析. データサイエンティスト育成講座：東京大学の：Python で手を動かして学ぶデータ分析. マイナビ出版, 2019.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. Vol. 16, pp. 321–357.

[20] 竹村彰通, 小川光紀. 異常値検知のための基本的ﾓﾃﾞﾙの考察冊. 国立研究開発法人　宇宙航空開発研究機構,

2014.

[21] Miki Kurihara, Masahiro Tsujimoto, Megan Eckart, Caroline A. Kilbourne, Frederick Matsuda, Brian J. McLaughlin, Shugo Oguri, Frederick S. Porter, Yoh Takei, and Yoichi Kochibe. Ground test results of the electromagnetic interference in the x-ray microcalorimeter onboard XRISM. In *Space Telescopes and Instrumentation 2022: Ultraviolet to Gamma Ray*, Vol. 12181, pp. 1445–1458. SPIE.

[22] Ryuta Imamura, Masahiro Tsujimoto, Hisamitsu Awaki, Meng P. Chiao, Ryuichi Fujimoto, Yoshitaka Ishisaki, Richard L. Kelley, Caroline A. Kilbourne, Frederick S. Porter, Makoto Sawada, Gary A. Sneiderman, Yoh Takei, and Shinya Yamada. Results of accelerometer monitor in the ground testing of resolve x-ray microcalorimeter instrument onboard XRISM. In *X-Ray, Optical, and Infrared Detectors for Astronomy X*, Vol. 12191, pp. 763–770. SPIE.

[23] Peter J. Shirron, Mark O. Kimball, Bryan L. James, Theodore Muench, Edgar R. Canavan, Michael J. DiPirro, Thomas G. Bialas, Gary A. Sneiderman, Kevin R. Boyce, Caroline A. Kilbourne, Frederick S. Porter, Richard L. Kelley, Ryuichi Fujimoto, Yoh Takei, Seiji Yoshida, and Kazuhisa Mitsuda. Design and on-orbit operation of the adiabatic demagnetization refrigerator on the hitomi soft x-ray spectrometer instrument. In Jan-Willem A. den Herder, Tadayuki Takahashi, and Marshall Bautz, editors, *Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray*, Vol. 9905, p. 99053O. International Society for Optics and Photonics. Issue: 02.