

# 機械学習高速自動スペクトル解析ソフト”EMPeaks”について

永村 直佳<sup>1,2\*</sup>・安藤 康伸<sup>3</sup>

<sup>1</sup>物質・材料研究機構 マテリアル基盤研究センター 〒305-0003 茨城県つくば市桜 3-13

<sup>2</sup>東京理科大学大学院 先進工学研究科 〒125-8585 東京都葛飾区新宿 6-3-1

<sup>3</sup>東京工業大学 科学技術創成研究院 化学生命科学研究所 〒226-8501 横浜市緑区長津田町 4259 R1 棟 R1-26

(20XX年XX月XX日受付；20XX年XX月XX日掲載決定)

## Introduction for “EMPeaks”, high-throughput spectrum analysis assisted by machine learning

Naoka Nagamura<sup>1,2</sup> and Yasunobu Ando<sup>3</sup>

<sup>1</sup>Center for Basic Research on Materials, National Institute for Materials Science, 3-13, Sakura, Tsukuba, Ibaraki 305-0003

<sup>2</sup>Graduate School of Advanced Engineering, Tokyo University of Science, 6-3-1 Nijjuku, Katsushika, Tokyo 125-8585

<sup>3</sup>Institute of Innovative Research, Tokyo Institute of Technology, 4259, Nagatsuta, Midori, Yokohama, Kanagawa 226-8501

(Received XXX XX, 20XX ; Accepted XXX XX, 20XX)

There are a wide variety of measurement techniques that produce spectra as output datasets. Existing data analysis software and some open-source macros are useful but not sufficient for non-experts to perform peak-fitting analysis and interpretation. Therefore, we have investigated analysis method using unsupervised machine learning for high-throughput and automated peak deconvolution analysis of spectra without prior knowledge of experimental techniques and material databases. In this paper, we will introduce the open-source package called “EMPeaks”, including development chronology, specific usage, current issues, and analysis examples.

KEYWORDS: spectroscopy, peak fitting, data analysis, machine learning

### 1. はじめに

X線光電子分光(XPS)にラマン分光, X線吸収・発光分光, 電子エネルギー損失分光(EELS)に走査型トンネル分光…と, スペクトルが出力される表面科学の計測手法には枚挙にいとまがない。機器付属ソフトや汎用解析マクロが普及してはいるものの, スペクトル解析には測定対象はもとより装置のクセも含めた深い前提知識を要し, 匠の技と揶揄されることもある。

そこで筆者らは機械学習を活用して, 前提知識なしでもスペクトル解析を精度よく, 自動的かつ高速で行える, 属人性を低減した解析手法を検討してきた<sup>1-4)</sup>。本稿では, その出口として開発した解析パッケージ”EMPeaks”について, 物理的考察と実験データでの動作検証を担う永村と, アルゴリズム開発とプログラム実装を担う安藤の対話形式で紹介する。

### 2. 開発の経緯

**永村:** 機械学習を援用してスペクトルデータ解析をやってみたいと考えた発端は, 放射光を使った顕微分光計測で, マッピング計測に伴う大量の XPS スペクトルを扱う必要に迫られた時でした。ラボで数時間かけてためた数本のスペクトルを解析する分には研究室秘伝の解析マクロや解析ソフトで十分でしたが…。

**安藤:** Materials Genome Initiative(米, 2011)に始まり, 日本でも新学術領域研究, JST イノベーションハブ構築支援事業(MI<sup>2</sup>I), 戦略的イノベーション創造プログラムなどを通して, インフォマティクスの知見を活用した材料研究基盤が急速に整備されてきた中で, 各種データ解析にも機械学習を活用しよう, という流れも強まっていました。

**永村:** 計測データには, ピーク位置や半値幅は勿論, 成分数も異なる多様な形状のスペクトルが含まれます。スペクトルの成分分離は, 関数形と成分数を予め

\*E-mail: NAGAMURA.Naoka@nims.go.jp

決め打ちして、非線形最小二乗法を用い、ピーク位置と半値幅に適切な初期値を設定してカーブフィッティングする、という手順が一般的です。この手順を実行する解析マクロで自動解析を走らせることも可能ですが、初期値はえいやっと決めてスタートするので、自動解析終了時のデータを確認してみると、局所解に落ちて殆どのスペクトルがまともにカーブフィッティングできていない、という事態もありまして…。成分数も適切な初期値もバラバラなスペクトルが含まれるデータセットに対して「いい塩梅に一括処理する」には、人工知能でも使わないと厳しいな、と。

**安藤**：私は表面・界面の物性理論が専門ですが、固液界面の分子動力学シミュレーションの結果から化学反応を起こしている分子を発見する方法など、手間のかかるデータ解析に機械学習を応用したいと考えていました。また、第一原理計算の枠組みでは難しい大規模計算や材料スクリーニングをサポートするための機械学習ポテンシャルも研究・利用しています。このような私の問題意識と、大量のスペクトルデータを自動解析して、解釈しやすいように情報抽出したい、という永村さんの興味の方向性は一致していました。

**永村**：私自身は分光分析実験が専門で、情報科学は素人でしたが、走査透過型顕微鏡(STEM)-EELS のスペクトルイメージングデータに対して、非負値行列因子分解で解析することで情報抽出できる、という話<sup>5,6)</sup>を聞いて、がぜん機械学習の活用に現実味を感じました。そこから機械学習の入門本<sup>7,9)</sup>で少し勉強して、既に機械学習を扱っていた安藤さんに相談しました。

**安藤**：はい、この相談を受け、どんな課題に対しても万能な機械学習アルゴリズムやモデルは存在しないので(ノーフリーランチ定理)、個別の課題に対して最適なアルゴリズムを開発するアプローチで研究に着手しました。インフォマティクスの活用にあたっては、課題設定と目的の明確化が最も重要です。

**永村**：今回の場合は、ノイズがのった実験データに対して、実用に足る精度を保ちつつ、高速に(=低計算コストで)、XPS 内殻スペクトルを成分分離する、という目的ですね。

**安藤**：スペクトル 1 本に対して、最適な成分数の推定も含めて、成分分離解析の精度を徹底的に追及する、という目的だと、ベイズ推定を使う取り組み<sup>10,11)</sup>があります。ただし、計算コストは気にしていない。

**永村**：顕微分光では大量のデータや画像データが得られるので、解析と機械学習の相性はよいですね。顕微鏡の画像処理などは産学問わず AI 活用が進められて

いる印象です。

### 3. 利用方法

**永村**：それでは、表題のスペクトル解析ソフト「EMPeaks」について解説していきましょう。

**安藤**：我々の提案する解析手法では、スペクトルを確率密度分布関数のようなものとみなします。確率密度分布のパラメータを最尤推定する手法の一つに Expectation-Maximization(EM)アルゴリズムがよく知られていて、ニュートン法のような反復法です。通常の EM アルゴリズムはイベント列に関する 1 次元データしか扱えないために放射光データ解析には計算が重すぎたため、スペクトルの強度を重みとしたヒストグラム形式のデータを扱うことができる「スペクトル解析に適した EM アルゴリズム」を考案しました。

**永村**：EM アルゴリズムは意外と系を選びますよね。各成分が Gauss 分布だとよいのですが、XPS の成分分離解析でよく使うような Voigt 関数や Doniac-Šunjić 関数の混合分布関数だと計算量が増えてしまったり、収束が悪くなったり…。

**安藤**：そこは、EM アルゴリズムをより一般化して、多変数関数の最適化問題に用いられる座標降下法を取り入れた ECM アルゴリズムを適用することでカバーできています！現時点でもいくつかの種類のバックグラウンド処理も可能になっていますよ。ランダムな初期値を与えても安定に動作しますし、初期値を振って最良のモデルを探すことも容易です。

**永村**：アルゴリズムの詳細は表面と真空の過去の解説記事<sup>1)</sup>などをご参照いただくとして、実際に EMPeaks を使ってみましょう。

**安藤**：EMPeaks は R で開発を進めていますが、利用しやすいように Python でもパッケージ化しています。ソフトウェアリポジトリの Python Package Index(PyPI)にアップロードしていますので、こちら<sup>12)</sup>から無料でお試しください。

**永村**：Python の環境は人それぞれですが、私は Windows ユーザーで、ずっと Anaconda を使っていました。ただ conda と pip のインストールがうっかり混在すると環境破壊が起きるので、最近 Visual Studio Code(VS Code)をエディタとして Jupyter Notebook でコードを書き、pip でパッケージを入れています。また、各種パッケージの動作に Python のバージョン依存性があることもあるので、pipenv を使って仮想環境を構築するようにしています。

```
PS C:\Users\...\(作業フォルダ)...\Analysis> pipenv -python 3.11
Creating a virtualenv for this project...
(仮想環境立ち上げ中)
Successfully created virtual environment!
Virtualenv location: C:\Users\...\(作業フォルダ)...\Analysis\.venv
Creating a Pipfile for this project...
PS C:\Users\...\(作業フォルダ)...\Analysis> pip install EMPeaks
Installing EMPeaks...
Resolving EMPeaks...
Added empeaks to Pipfile's [packages] ...
Installation Succeeded
...
PS C:\Users\...\(作業フォルダ)...\EMPeaks2024>
```

**Fig.1.** (color online) Schematic image of the terminal window in VS Code at the first step.

**安藤：** Anaconda を用いても Navigator 上で仮想環境を構築できます。Python を単一の環境だけで扱っていると、別のパッケージを導入した際に対応バージョンの矛盾が生じるなど、トラブルの原因になります。また、トラブルで全ての環境が召されてしまうと全て再構築しなければいけなくなります。ですので、研究プロジェクトごとに仮想環境を構築することをお勧めします。代表的な仮想環境の構築では Anaconda, pipenv, venv, Windows Subsystem for Linux 2(WSL2)などが利用可能です。なお Anaconda は商用利用ライセンスが有料化されましたので、企業・国立研究所の方はライセンスを購入するか、Miniconda をご利用下さい。

**永村：** ではコマンドラインを見ていきましょう。仮想環境内で Python のバージョンを指定して実行し、EMPeaks をインストールします(Fig.1)。以前、職場のネット環境で pip(pipenv) install EMPeaks を試したらプロキシエラーではじかれましたが(笑)。

**安藤：** 回避方法は色々あると思いますが、PyPI から EMPeaks ライブラリをダウンロードして、オフラインでインストールする手がありますね。例えば、「ファイルをダウンロード」の Source Distribution にある圧縮ファイルをダウンロードして解凍し、PC のモジュール置き場に移してもらって、Jupyter Notebook からモジュール置き場のフルパスを指定して読み込むとか(Fig.2 の1つめのセル)。また、ダウンロードしたファイルから pip でインストールすることもできます。

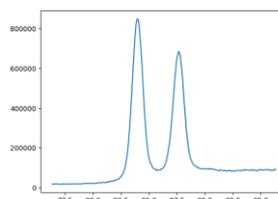
**永村：** 作業フォルダ(モジュール置き場と同じ場所だと安全)に Notebook ファイルと解析したいスペクトルデータが記録されているファイル(Au4f.dat)をおきます。1 行目にヘッダーがついていて、1 列目に昇順でエネルギー、2 列目に強度(カウント数)が記録されているスペクトル 1 本の情報が含まれるデータ(Fig.3(a))を読み出し、Gaussian でフィッティングしてみます。

**安藤：** チュートリアルサイト<sup>13)</sup>にあるサンプル Notebook のうち、実測データのフィッティングに関する

```
# EMPeaksをインポートする
import sys
sys.path.append(R"C:\Users\...(EMPeaksが入っているフォルダ)...\EMPeaks2024")

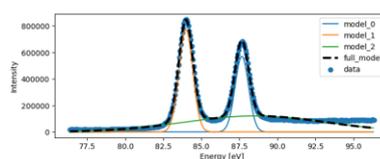
# 他に必要なパッケージをインポートする
%matplotlib inline
from EMPeaks import GaussianMixture
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

Au4f = pd.read_csv('Au4f.txt', delimiter='t')
plt.plot(Au4f['BE'], Au4f['Int'])
# Kは決め打ちするピークの本数。
# ここでは便宜上バックグラウンドを1つのピークとみなしています。
gmm = GaussianMixture.GaussianMixtureModel(K=3)
gmm.N_tot = sum(Au4f['Int'])
gmm.plot(Au4f['BE'], Au4f['Int'])
```



```
# 5回の初期パラメータサンプリングのあとに得られた最適なモデルをプロットします。
gmm.sampling(Au4f['BE'], Au4f['Int'], trial=5, r_epsilon=1e-8)
gmm.plot(Au4f['BE'], Au4f['Int'])
```

```
Starting Trial # 0
<< Start fitting via Adapted EM Algorithm. >>
Convergence is achieved at iter. 73, elapsed time 0.28 s
...
Sampling the different initial guess with 5 trial is finished.
Maximum Log-Likelihood is obtained in trial 3
Best model parameters and scores of samples are following:
mu: 87.661 eV 83.981 eV 88.530 eV
sigma: 4.428e-01 4.562e-01 4.668e+00
N_tot: 2.932e+06
N: 6.338e+05 8.839e+05 1.414e+06
pi: 2.162e-01 3.015e-01 4.823e-01
LL: -1.44814140e+08
RMSE: 1.84392889e+04
```



**Fig.2.** (color online) Schematic image of the program window at the fitting calculation using Gaussian.

(a)	BE	Int	(b)	Wave	Int	
	76.35	17865		300.1541		19.75
	76.4	18368		301.32	27.5	
	76.45	17780		302.4857		23
	76.5	19111		303.6512		19.75
	76.55	16924		304.8164		28
	76.6	18183		305.9814		32.75
	76.65	18743		307.1461		24.5
	76.7	18950		308.3106		21.25
	76.75	19563		309.4748		22.75
	76.8	19120		310.6388		23.5
	76.85	17251		311.8025		19.25
	76.9	18166		312.9659		21
	76.95	17887		314.1292		17.75
	77	17435		315.2921		22.75
	...	...		...		...

**Fig.3.** Input data for EMPeaks. (a) shows the contents of the “Au4f.dat” and (b) of the “Raman.dat”.

る部分を抜粋して実行していきます。

**永村：** コマンドラインを示す図の文字が小さいのはご容赦下さい。サンプル Notebook から内容を確認できます。Fig.2 の 2 つめのセルで必要なパッケージを入れ、3 つ目のセルでデータを読み込み、成分を決め打ちで入力します。バックグラウンド処理については後述しますが、ここではさしあたりバックグラウンドを幅が大きな 1 つの Gaussian とみなしましょう。Fig.2 では成分数は  $K=3$  と設定して計算を進めています。

**安藤：** 成分数を推定するアルゴリズムについても検討しています<sup>14)</sup>。

**永村：** 局所解へのトラップを避けるため、初期パラメータを何種類か適当にサンプリングしてから、得られた最適なモデルをプロットします。4 つ目のセルを実行すると無事にフィッティングができて、各ピーク成分のパラメータ(ピーク値, 分散, 強度など)の結果が出力されます。Au 4f と言いつつ、2 つのピーク(4f<sub>7/2</sub> と 4f<sub>5/2</sub>)の強度比(3:4 になるはず)やスピン軌道相互作用で決まる分裂幅などは考慮せず、独立した 2 つのピーク成分として扱っていることはご容赦下さい…。

**安藤：** ECM アルゴリズムを使った Gaussian 以外の関数形でのフィッティングも全く同様にできます。EMPeaks では Lorentzian, Pseudo Voigt 関数と Doniach-Šunjić 関数を用意しています。

**永村：** Igor Pro や OriginPro などを使って後でグラフ化できるようにフィッティング結果を csv データで保存する、元データのエネルギーが昇順になっていない場合に対応する、1 つのデータファイルに複数本のスペクトルの情報が含まれている場合に対応する、といった機能を含むチュートリアルファイルはこちら<sup>13,14)</sup>にありますので、ご参照ください。

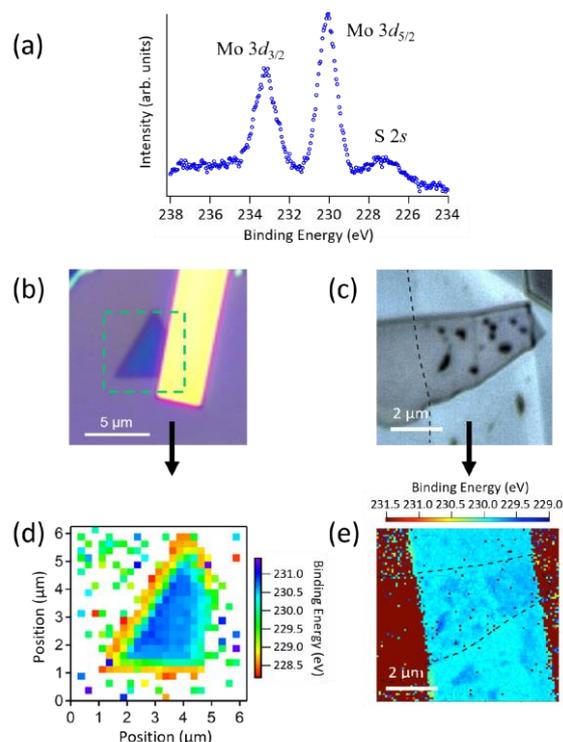
**安藤：** そのための動機が大量のスペクトルを含むデータセットの自動フィッティングですもんね。

**永村：** そうなんです。上記のチュートリアルファイルでは、私が使っていた顕微 XPS 装置の出力形式に合わせて、ヘッダーやインデックスは無しで、1 行目にエネルギー、n 行目に n-1 本目のスペクトルの強度(カウント数)、というデータファイルを読み込む形式になっています。お手持ちのデータセットに合うようにカスタマイズしていただく必要はあります。

**安藤：** 放射光顕微 XPS のデータを EMPeaks で解析して、結合エネルギーのピークシフトをマッピングした例が Fig.4 です。

**永村：** これは EMPeaks ありきの解析結果でした。数百～数千本のスペクトルデータが含まれていますから。

**安藤：** EMPeaks の ver 2.1 からは、一定値、線形、単調

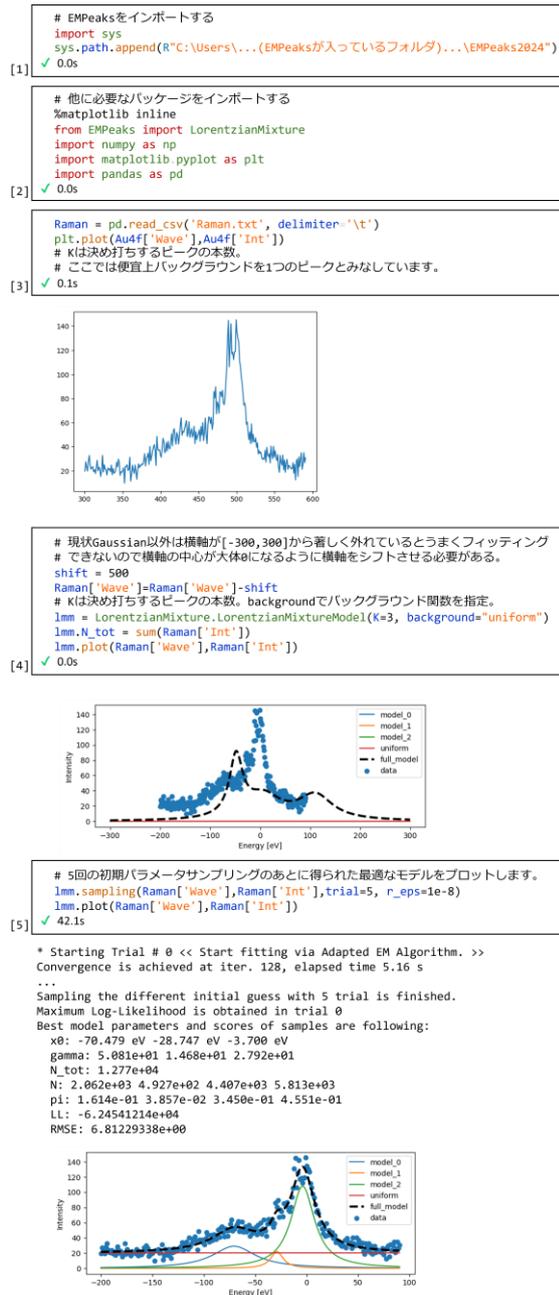


**Fig.4.** (color online) Examples of analysis results in scanning photoelectron microscopy (SPEM) using EMPeaks<sup>14,16)</sup>. (a) Typical XPS spectrum of Mo 3d and S 2s core levels at MoS<sub>2</sub> atomic layer sheets. (b,c) Optical microscope images of (b) a monolayer MoS<sub>2</sub> flake with Nb doped edge<sup>15)</sup> and (c) an all 2D heterostructure tunnel field-effect transistor composed of MoS<sub>2</sub>, WSe<sub>2</sub>, and hexagonal BN atomic layer flakes<sup>14,16)</sup>. (d,e) Photoelectron peak-position mapping images of Mo 3d<sub>5/2</sub> obtained by SPEM and analyzed by EMPeaks. Dot-lines are inserted to recognize the domains.

増加な折れ線状、の 3 種類の関数形でバックグラウンド処理と上記のピークフィッティングを同時に実施できるようになっています。

**永村：** バックグラウンド処理については私も熱望していました！

**安藤：** 1 列目に昇順でエネルギー値、2 列目に強度(カウント数)が記録されているデータ(Raman.dat)のスペクトルを Lorentzian でフィッティングした時の実行画面が Fig. 5 です。4 つ目のセルで lmm (Lorentzian Mixture Model という意味)という関数を定義するところで、引数として、成分数の  $K$  に加えて background でバックグラウンド関数を指定します。一定値の場合は「background="uniform"」、線形の場合は「background="linear"」、折れ線状の場合は全エネルギー領域における折れ線の分割数を  $n$  個にしたい場合「k\_ramp=n-2, background="ramp\_sum"」として下さい。ただし現状、折れ線モデルは単調増加に限ります。



**Fig.5.** (color online) Schematic image of the program window at the fitting calculation using Lorentzian.

**永村:** Fig. 2 や Fig. 5 の計算は私のノート PC (CPU: Intel(R) Core(TM) i7, 2.1 GHz, メモリ: 16 GB) で実行しています。1 本のスペクトルに対する計算にかかった時間は, Fig. 2 の Gaussian で 2.2 秒, Fig. 5 の Lorentzian で 42.1 秒となっています。1 万本のスペクトルを処理する場合, Gaussian だと 1 本あたり 2 秒として, 2 万秒で 5 時間半程度。マッピング計測にかかる時間と同程度なので, 実験しながら解析して, その結果をふまえて次の測定を... というサイクルが可能になります。なにより, 局所解に落ちる心配をせず計算を任せて,

他の作業をしたり休憩できる, というメリットは非常に大きいです。実験中は解析のことばかり考えているわけにはいかないのです。

**安藤:** Lorentzian だとこの 20 倍かかっているのです, このままだとリアルタイム解析は難しそうです。ただ, 最近では放射光施設で扱う実験データが膨大化しているので, データの圧縮, 管理, 解析を補助するデータセンターを整備する動きが加速しています。実装言語の変更でもかなり高速化が可能ですし, 解析処理の並列計算も実施しやすくなると期待しています。

**永村:** Python はライブラリが充実していて初心者に優しいのですが, 実行速度が遅いと聞きますね。将来的には EMPeaks を放射光施設の PC に標準解析ツールとして組み込む野望も! ?

**安藤:** XPS に限らず X 線吸収分光法や蛍光 X 線分光法など様々なスペクトル解析に活用できるポテンシャルはありますし, 放射光実験の進め方を変えられるのではないかと期待しています。

**永村:** 実際に我々の研究室では, 顕微ラマン分光のデータ解析に活用して, 原子層材料の層数判定やキャリア密度分布評価に活用しています。あと, EMPeaks はスペクトルに限らず分布関数が出力されるタイプのデータ解析に適用可能なので, 反射高速電子回折 (RHEED) の画像を輝度ヒストグラムにして, 横軸輝度, 縦軸ピクセル数の分布関数をピークフィッティングして特徴を可視化する, という取り組みも行っています。「スペクトルや分布関数は手元にデータとして沢山あるけど細かい解析が面倒で塩漬けされている」という方は, 是非 EMPeaks を試してみたいです。

#### 4. 現状における課題

**安藤:** EMPeaks の利用法を紹介してきましたが, まだ色々課題があることは事実です。

**永村:** XPS を日頃から扱っていると, バックグラウンドとして Shirley 法や Tougaard 法を使うことも多いのですが, これらを EMPeaks で扱うのは難しいですか?

**安藤:** はい, テクニカルに難しいというのが正直なところです。その代わりに, シグモイド型のステップ関数や長周期のバックグラウンドに対するモデリングを開発中です。

**永村:** それから, ピーク成分の関数に関しても, Pseudo-Voigt 関数ではなくて Voigt 関数を使う人が多いのかなと思います。ただ, こちらも計算量が膨大になってしまうんですね?

**安藤:** Lorentzian の分散を決め打ちしておけば計算量を

減らせるので、実装できる可能性はあるかなと。

**永村**：そこは粒子寿命のパラメータで試料によって決まるので、決め打ちでもいいかなという気はします。ただ、データ解析の属人性低減という観点では前提知識を要するパラメータを入力させるのは微妙ですが。

**安藤**：属人性低減の試みに関して、やはり成分数も自動で推定してほしいという声は多くいただいています。計算コストの低さはキープしつつ、スパースモデリングによる成分数推定のアルゴリズムも検討して論文化したばかりです<sup>14)</sup>。今後 EMPeaks の方でも実装できたらと考えています。

**永村**：最後に最も悩ましい課題ですが、デバッグ・メンテナンス・技術相談に人員と時間を割くのがなかなか難しい。研究者としての仕事は、アルゴリズム自体の開発や、アルゴリズムの新しい応用例の発見、といったテーマで論文化できるところまで。せっかく有益なパッケージを開発しても、普及させるための作業はなかなか評価されませんからね…。

**安藤**：このパッケージがデータ解析に必須となる研究テーマで予算化できれば、システム開発業者に外注するのもよいかと。また、現状オープンソースですので、企業さんも含めてユーザーが増えていけば、コンソーシアム化して、共同管理や修正パッケージ・拡張パッケージの共有を行っていく方向性もあると思います。

## 5. まとめ

本稿では、機械学習を活用してスペクトルのピークフィッティング処理を行うパッケージ“EMPeaks”について紹介した。これは著者らがイメージング計測などで得られるスペクトルビッグデータからの効率的な情報抽出の必要性に駆られて開発を進めているものであり、解析処理の自動化、高精度化(局所解への落ちにくさ)、計算コストの低減、属人性の低減、といった解析処理一般における課題解決に挑むものである。

便利なツールとして AI 技術が急速に普及し、データ解析にも多種多様なアプローチが日々台頭しており、ここ数年はさながら群雄割拠の AI 活用戦国時代である。先端計測もその流れにのって、計測・解析の超効率化が進み、データから研究者が見逃していた重要な知見を AI が導き出す未来が期待される。

## 6. 謝辞

本稿で紹介した研究およびパッケージ開発は、JST さきがけ(JPMJPR17NB, JPMJPR20T7)、JST CREST

(JPMJCR1761, JPMJCR21O1)、NEDO(P16010)、防衛装備庁安全保障技術研究推進制度(JPJ004596)の支援を受けて実施した。本パッケージの実装には、松村太郎次郎氏(産業技術総合研究所, AIST)が多大に貢献している。アルゴリズム開発に関しては、永田賢二氏(NIMS)、赤穂昭太郎氏(AIST)と進めた。拡張パッケージ作成は福健太郎氏(東京理科大学博士研究員)、吉成朝子氏(同博士課程)、後藤陸氏(同修士課程)と行った。

## 文 献

- 1) 永村直佳, 松村太郎, 永田賢二, 赤穂昭太郎, 安藤康伸 : 表面と真空, **64**, 382 (2021).
- 2) T. Matsumura, N. Nagamura, S. Akaho, K. Nagata, Y. Ando : Sci. Tech. Adv. Mater. (STAM) **20**, 733 (2019).
- 3) T. Matsumura, N. Nagamura, S. Akaho, K. Nagata, Y. Ando : STAM Methods **1**, 45 (2021).
- 4) T. Matsumura, N. Nagamura, S. Akaho, K. Nagata, Y. Ando : STAM Methods **3**, 2159753 (2023).
- 5) 武藤俊介, 志賀元紀 : 表面科学, **37**, 610 (2016).
- 6) M. Shiga, K. Tatsumi, S. Muto, K. Tsuda, Y. Yamamoto, T. Mori, T. Tanji : Ultramicroscopy **170**, 43 (2016).
- 7) 須山敦志, 杉山将 : “ベイズ推定による機械学習入門” (講談社, 2017).
- 8) A. C. Muller(著), S. Guido(著), 中田秀基(訳) : “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” (オライリージャパン, 2017).
- 9) C. M. Bishop(著), 元田浩(訳) : “パターン認識と機械学習(上・下)” (丸善出版, 2012).
- 10) A. Machida, K. Nagata, R. Murakami, H. Shinotsuka, H. Shouno, H. Yoshikawa, M. Okada : STAM Methods **1**, 123 (2021).
- 11) Y. Mototake, M. Mizumaki, I. Akai, M. Okada : J. Phys. Soc. Jpn. **88**, 034004 (2019).
- 12) EMPeaks (PyPI) <https://pypi.org/project/EMPeaks/>
- 13) 安藤の異動(2024-)に伴い、2024年8月現在チュートリアルサイトは移行中。現状の関連リンクは <https://researchmap.jp/naganao/%E8%B3%87%E6%96%99%E5%85%AC%E9%96%8B> または [http://www8.plala.or.jp/naganao\\_surface/EMPeaks.html](http://www8.plala.or.jp/naganao_surface/EMPeaks.html)
- 14) T. Matsumura, N. Nagamura, S. Akaho, K. Nagata, Y. Ando : STAM Methods **4**, 2373046 (2024).
- 15) M. Okada, N. Nagamura, T. Matsumura, Y. Ando, A. K. A. Lu, N. Okada, W-H. Chang, T. Nakanishi, T. Shimizu, T. Kubo, T. Irisawa, T. Yamada : APL Mater. **9**, 121115 (2021).
- 16) K. Nakamura, N. Nagamura, K. Ueno, T. Taniguchi, K. Watanabe, K. Nagashio : ACS Appl. Mater. Interfaces **12**, 51598 (2020).