python による機械学習を fortran 世代がやってみた

榊原 昇一、松崎 剛
(大阪大学 産業科学研究所)
shouichi@sanken.osaka-u.ac.jp, matsuzaki@sanken.osaka-u.ac.jp

1. はじめに

昨年2019年、産業科学研究所に AI センターが設置されました。AI センターが開催した全職員向けの勉強会や報告会を通じ、AI の様々な活用方法を聴く機会に恵まれました。その中で化学反応の最適条件をガウス過程回帰によって求めた講演を聴き、その講演内容はすでに論文として公開されているものでした[1]。設定するべきパラメーターがいくつかある中で最適な条件を求めることは工作でも計測でもありとあらゆる分野で必要になります。講演内容の論文を参考にし、燃焼法による元素分析の最適な設定をガウス過程回帰で求めた試みについて本報告では書いてゆきます。

2. ガウス過程回帰

ガウス過程回帰という言葉は「ガウス過程」と「回帰」という言葉から成り立っています。「回帰」はある入力に対して出力を予測することで、実験データをグラフにする時に何らかの回帰曲線をプロットすることがよくあります。その際、背後にある現象を探るために何らかの関数でフィッティングすることが多いと思います。例えばある時定数で物質 A が物質 B に変わる時、A か B の量を測定してやり指数関数でプロットすれば時定数が分かります。これが一つの時定数プロットに乗らない場合は B がさらに変化していることなどが予想されてくるわけです。

回帰を行う際、その背景にある現象が予想されていれば基底関数を選んで重みづけを行い、和を取ってやれば実行できます。

$$\hat{y} = w_0 + w_1\phi_1(x) + w_2\phi_2(x) + \cdots + w_H\phi_H(x) \qquad (1)$$

ここで $\hat{y}$ は回帰曲線を表し、w は重み、ϕ は基底関数を表します。この回帰の表現は重みについて線形という意味で線形回帰と呼ばれます(よくみかけるフィッティングは線形回帰だと思います)。入力 x に対する出力 y について N 個のセットを取得した際、H 個の重みを求めるには、回帰曲線と各データとの垂直距離の二乗の和が最小になる条件を用います。つまり最小二乗法です。その結果は行列で書くとスッキリして、次の線形代数の問題を解くことになります。

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{y} \ .$$

ここで、

$$\boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_H \end{pmatrix} \quad \boldsymbol{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix} \quad \boldsymbol{\Phi} = \begin{pmatrix} 1 & \phi_1(x_1) & \cdots & \phi_H(x_1) \\ 1 & \phi_1(x_2) & \cdots & \phi_H(x_2) \\ \vdots & & & \vdots \\ 1 & \phi_1(x_N) & \cdots & \phi_H(x_N) \end{pmatrix}$$

を表します。したがって H 個の重みを求める計算は H x H 行の行列 $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ の逆行列を求める計算が主となります。

さて、背景がわかっている現象を回帰する場合は基底関数があらかじめ推測でき、線形回帰を使うことができますが、一般的な問題にあたっては基底をたくさん用意する必要があるでしょう。また入力は 1 次元とは限りません。入力の次元数が増えると求めるべき重みも次元数の冪乗で増えてゆき処理できなくなります。そこで物の考え方を改めて、「現実世界における観測 y は何らかの確率分布 p(y) からのサンプリングによって得られたものだ」と考える仮説を導

入します。これを確率的生成モデルといいます[2]。この話、すんなり理解するのは難しいのですが、例えばサイコロを N 回振って出た目の平均を計算する場合、出た目の合計を試行回数 N で割りますよね。N を増やすとその平均が 3.5 に近づいて行くことを想像する方は、サイコロの目が均等に 6 分の1の確率で出ると考えているわけです。つまり均等な確率分布を意識しているわけです。何らかのデータを取得するのはその背後にある現象の一部を切り取ってきている訳で、平均は和を割って表されますが、ある変数が起こる確率が与えられた現象の場合は、平均は変数と確率を掛けたものの和や積分になります。私がイメージしている確率的生成モデルとはこのように何らかの分布を持ったデータの一部を私たちが見ていると理解することです。そう考えると自然な解釈かと思います。

重みの数が増えてゆき処理できない事を回避するために、重みが平均ゼロで共分散行列が単位行列に比例($\lambda^2 I$)する多変量ガウス分布から確率的生成されたと考えてみます(分布の平均をゼロと置いて考えても、最後に原点をずらせばよいので普遍的な話ができます。共分散行列が単位行列に比例するという事は、それぞれの重み成分の間に相関がない事を表していて線形回帰と同じ考え方です)。すると(1)式のように w を線形変換した出力 y も多変量ガウス分布に従い、平均ゼロ、共分散行列は

$$\boldsymbol{\Sigma} = \mathrm{E}[\boldsymbol{y}\boldsymbol{y}^T] - \mathrm{E}[\boldsymbol{y}]\mathrm{E}[\boldsymbol{y}]^T = \mathrm{E}[\boldsymbol{\Phi}\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{\Phi}^T] = \lambda^2\boldsymbol{\Phi}\boldsymbol{\Phi}^T$$

となります。ここで E は期待値を取る演算を表し、確率変数でなく定数で構成されている Φ は演算の外に出ます。すると重み w は期待値計算の中で単位行列となり無くなります。つまり数が多すぎて処理できなくなる重みが計算から消えて無くなることになります。「いやそうは言っても基底関数の行列が残っているので計算が大変では？」となりますが、基底関数が無限個あっても行列 $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$ は N x N 行列となり計算処理が可能となります。結局重み w の分布を仮定することによりデータ y も上記共分散行列に従う多変量ガウス分布となります。そしてこのように、出力 y が多変量ガウス分布に従うとき、出力はガウス過程に従うと言います。実際の計算では基底関数も明示せず、滑らかな関数がサンプリングされる共分散行列(カーネル行列と呼ばれます)を導入します。

3. ガウス過程回帰の計算例

ここでガウス過程回帰を用いて最小値を探す問題を解いてみましょう。参考論文[1]に従い、python のガウス過程回帰モジュールとして GitHub で公開されている GPy を用いました。図1(a)の赤い曲線($\cos 2x + 0.12x$)が背後にあるとして、 0 ≦ x ≦ 10 の範囲内で最小値を探してみます。まず初期データとして x = 1, 3, 5, 7, 9 における出力を与えます(図1(a), (b)の黒×印。計算機にはこの点のみ見えています)。この 5 点を学習データとし、ガウス過程回帰を行なった結果が図1(b)の青線と青い領域です。青線は学習データを含んだ上で、データのない場所について未知とした多変量ガウス分布の期待値を表しています。青い領域は分布の幅±2σを表しており、信頼区間と呼ばれます。期待値のプロットは自在定規で引いたような曲線になっており、これと言った特徴的なところはありませんが、ガウス過程回帰の特徴は、分からない領域を分からないなりの幅で表してくれているところです。つまり信頼区間がデータのない場所では広く、データのある場所では狭くなっています。最小値を見つける問題ですので、信頼区間が低い場所まで伸びている点が候補に上がって来ます。そこで x = 0, 1.8, 4.5, 5.5, 10 を学習データに追加した計算結果が図1(c)です。境界の x = 0, 10 は最小点ではないことがわかり、どうやら 2 より少し少ないところに最小点がありそうだと見えてきました。背後にある曲線($\cos 2x + 0.12x$)も見えつつあります。最後に x = 1.7, 4.7, 8 を追加したのが図1(d)です。ここまでくると 2 より少し少ない場所に最小点があることが確信を持って言えると思います。さらにすごいのは期待値をプロットした青線が、背後にある曲線をほぼ再現していることです。このように出力を最小化・最大化する入力を探す際、ガウス過程回帰は強力なツールになります。

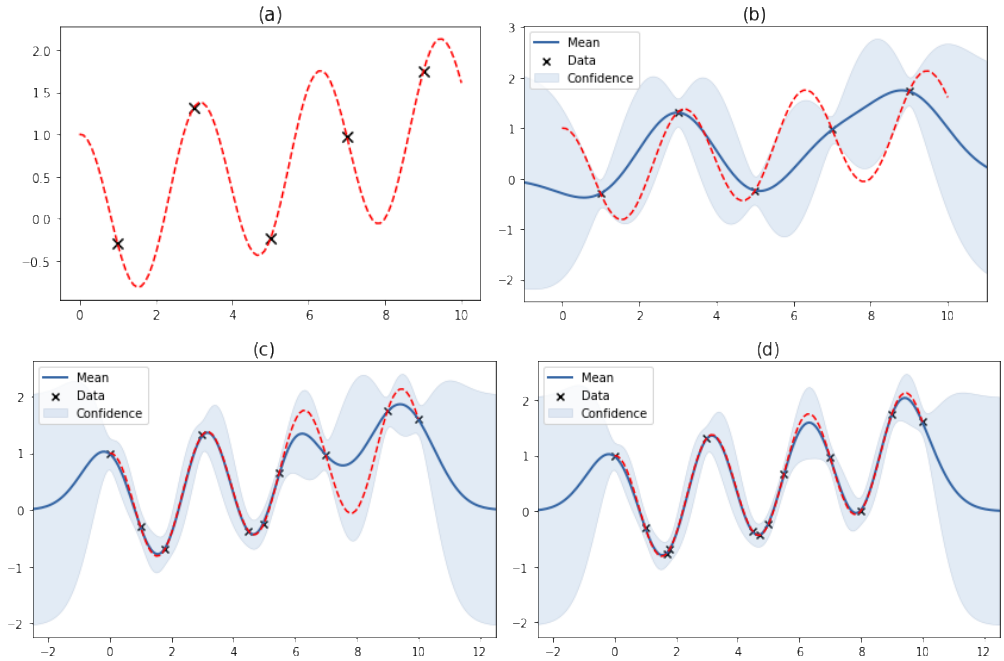

図1　ガウス過程回帰の例として未知の赤い曲線の最小点を探しています。黒×印のデータポイントを、回帰が示した最小点候補を選んで増やしています。

4. 元素分析条件の最適化

ここでガウス過程回帰を実際の問題に適用するにあたって、燃焼法による元素分析の最適条件を探してみましょう。対象となる化合物はメリット酸 $C_{12}H_6O_{12}$ でベンゼンの H がカルボキシル基で置き換わったものです。この化合物をメーカー推奨の条件で元素分析すると理論値からかなり外れた値が得られ、何らかの条件変更が必要となっていました（理論値からの差が0.3%あるといいデータとは言われないそうです）。表1にメーカー推奨の条件を、得られた測定結果を表2に示しました。

試料量（mg）	2
燃焼炉温度（℃）	850
酸素量（ml/min）	15
試料炉温度（℃）	950
He スイーピング時間（sec）	90

表1　元素分析のメーカー推奨条件

	実測された質量%	実測値/理論値
C	41.81（− 0.31 %）	0.9926
H	1.93（+ 0.15 %）	1.0904
N	0.00	1.00

表2　メリット酸の測定結果。理論値は C42.12%, H1.77%, N0.00%. 質量%のカッコ内の値は理論値からの差を表しています。

新しい測定条件を見つけるのにあたって、試料量は（節約して）1mg 固定としました。また変化量の大きい方が極値を求めやすいと予想して、実測値/理論値の比が 1 から大きく離れている水素にまず着目しました。その上で「燃焼炉温度と酸素量」、「試料炉温度と He スイーピング時間」の2条件2セットを振りながら最適条件を探してゆきます。

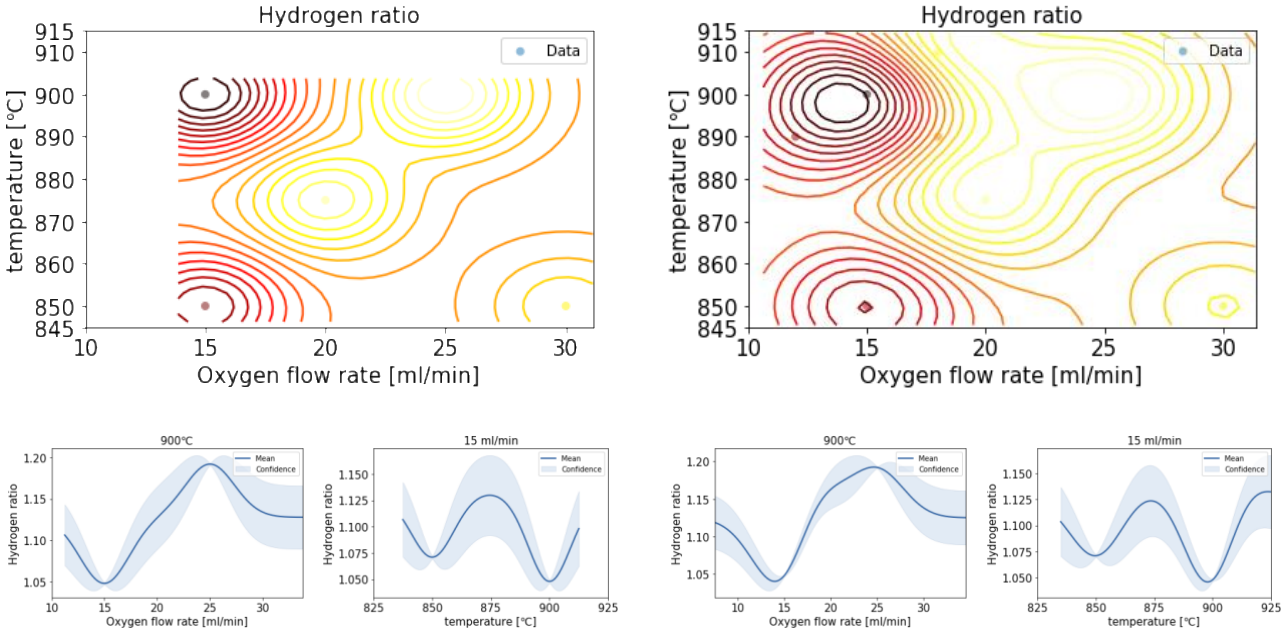

図2　酸素量と燃焼炉温度を振った時の水素実測値/理論値の比。上の図は 2 次元の等高線を表し、下の図は上の図の断面を表しています。

図3　図2の結果に3点追加した計算結果

計算1　試料炉温度 950℃、He スイーピング時間 90 秒固定とし、酸素量と燃焼炉温度を振る

［酸素量(ml/min), 燃焼炉温度(℃)] = [15, 850], [15, 900], [20, 875], [25, 900], [30, 850] の5点を学習データとし計算した結果が図2です。上の図は2変数プロットの等高線を表しており、色が濃い方が小さな値を表しています。左上と左下に極小点が現れています。下のふたつの図はそれぞれ燃焼炉温度 900℃で横方向に切った断面図と酸素量 15ml/min で縦方向に切った断面図を表しています。断面図のタイトルが断面を切った軸を表しています。予想される水素の実測値/理論値の比は 1 より大きいため、最小点を探してゆくことになります。これをみると燃焼炉温度 900℃の少し上下、酸素流量 15ml/min の少し上下に最小点の候補が見受けられます。

そこで新たに[12, 890], [18, 910], [18, 890] の3点を追加したものが図3です。信頼区間を考慮した最小点は、酸素量 15ml/min より少し下、燃焼炉温度 900℃ より少し下に存在しそうだと分かりました。そこで酸素量 14ml/min、燃焼炉温度 895℃ に条件を固定し、試料炉温度と He スイーピング時間の組み合わせを次に探ってみました。

計算2　酸素量 14ml/min、 燃焼炉温度 895℃ 固定とし、試料炉温度と He スイーピング時間を振る

［試料炉温度(℃), He スイーピング時間(秒)] = [910, 120], [930, 150], [950, 90], [970, 100], [990, 130], [1000, 140], [980, 120], [950, 140], [990, 110], [985, 125] の10点を学習データとし計算した結果を図4に示しました。これも計算1と同じように、最初の5点をまず計算し、そこから探るべき場所を見当付けながら点を増やしていった結果です。等高線をみてみると［試料炉温度, He スイーピング時間] = [950, 90] の計算1で用いた条件が極小値として下の方に現れている一方、[990, 130] 辺りにも極小値が現れている事が分かります。それぞれの断面を図4の下に並べました。この新しい極小値は気になるところですし、炭素の結果も見てみると[950, 90]辺りより、[990, 130]辺りが良い結果を与えていました。そこで断面図の信頼区間を考慮し、[試料炉温度(℃), He スイーピング時間(秒)] =[985, 125]

として再び酸素量と燃焼炉温度を振ってみました。

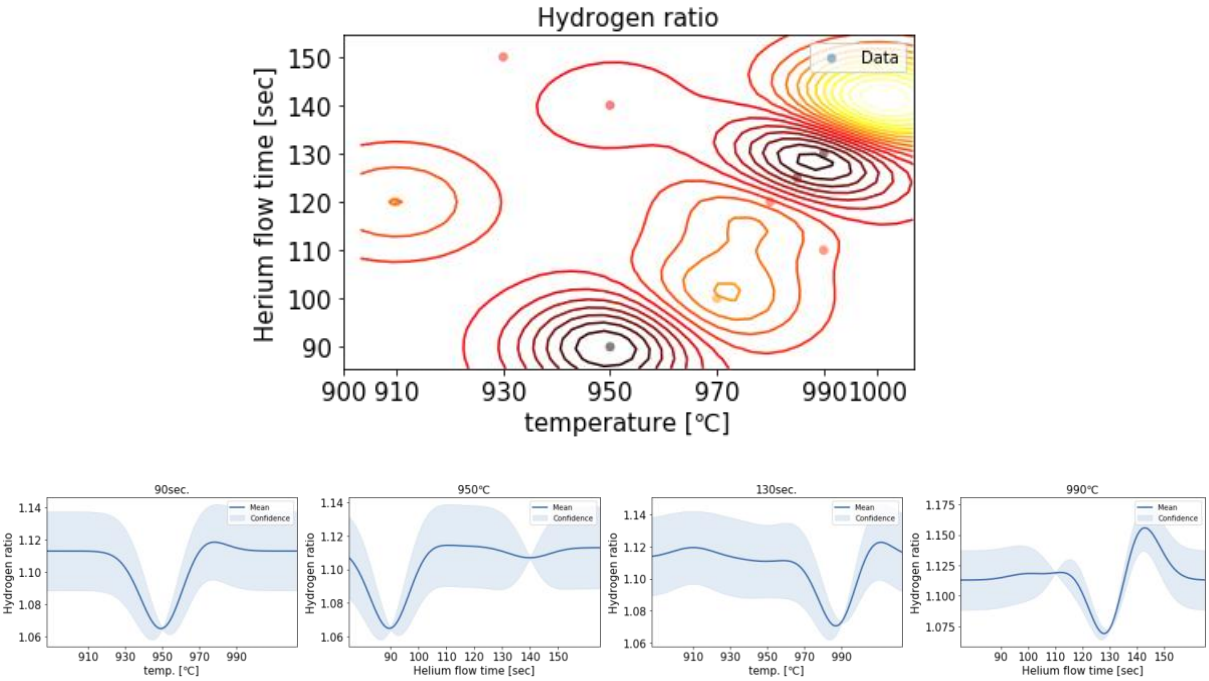

図4　試料炉温度とHe スイーピング時間を振った時の水素実測値/理論値の比

計算3　試料炉温度 985℃、He スイーピング時間 125 秒固定とし、酸素量と燃焼炉温度を振る

[酸素量(ml/min), 燃焼炉温度(℃)] = [15, 850], [15, 900], [20, 875], [25, 900], [30, 850], [14, 895], [17, 910], [25, 910], [20, 910], [17, 890], [25, 890], [20, 890] の12点を学習データとし計算した結果を図5に示しました。今回初めて、水素実測値/理論値の比が 1 を下回る領域が見えてきました。そして酸素量 15 - 25 ml/min, 燃焼炉温度 890 - 910 ℃ にかけて輪を描く領域が水素の実測値と理論値が等しくなることが分かりました。水素の測定条件として十分良い結果を生み出す条件が領域として現れてきたので、あとはこの領域内で炭素測定が良い場所を探せば良くなります。炭素の一番良かった条件を採用した条件と、最終結果を表3と4にまとめました。メーカー推奨設定でメリット酸を分析した表2と比較して、表4の結果はよく改善されていることが分かります。

試料量（mg）	1
燃焼炉温度（℃）	900
酸素量（ml/min）	15
試料炉温度（℃）	985
He スイーピング時間（sec）	125

表4　ガウス過程回帰で得られた試料量 1mg で分析した場合の最良の条件

	実測された質量%	実測値/理論値
C	41.99（- 0.13 %）	0.9969
H	1.82（+ 0.05 %）	1.0297
N	0.00	1.00

表5　表4の条件で得られたメリット酸の測定結果。理論値は C42.12%, H1.77%, N0.00%.質量%のカッコ内の値は理論値からの差を表しています。

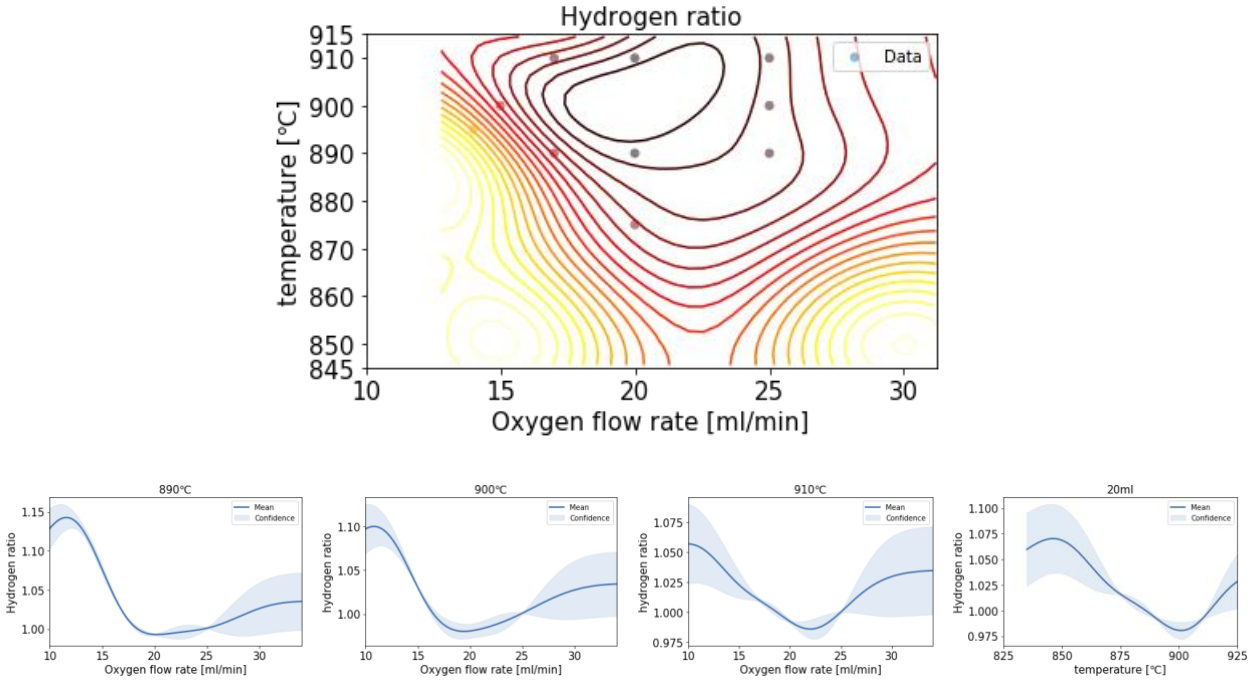

図5　酸素量と燃焼炉温度を振った時の水素実測値/理論値の比

5. まとめ

ガウス過程回帰について勉強し、GitHub で公開されている GPy モジュールを用いた実装を行いました。燃焼法による元素分析の条件設定に応用し、4 つのパラメーターをふたつずつ振ることにより、理論値との差が 0.03%以内に入る実測値を得られる条件を見いだすことができました。

この計算は 2015 年製の MacBook Air でも一瞬でできるものです。GPy の導入方法や、ガウス過程回帰そのものの解説記事もネット上に沢山あります。私も色々なサイトを読んで python を含めて今回勉強させてもらいました。どうぞ興味のある方は参考文献を読んでみてください。 特に2節の内容は参考文献[2]の第1章から第4章の内容を要約したものです。

参考文献

1) M. Kondo, et al, *Chem. Commun.,* 2020, 56, 1259.　(DOI: 10.1039/c9cc08526b)

2) 持橋 大地、大羽 成征 著 「ガウス過程と機械学習（機械学習プロフェッショナルシリーズ）」 講談社