

第28回数理科学コンクール課題解説

令和7年11月2日 千葉大学先進科学センター

目 次

1	はじめに	2
2	優秀者氏名	4
3	課題 1(対面・遠隔共通課題)	7
	解説	7
	講評	11
4	課題 2(対面・遠隔共通課題)	12
	解説	13
	講評	14
5	課題 3(対面・遠隔共通課題)	15
	解説	15
	講評	16
6	課題 4(対面・遠隔共通課題)	17
	解説	17
	講評	18
7	人工知能の部 課題	20
	解説	21
	講評	26

1 はじめに

明治の文明開化以来、我が国は欧米先進国の科学技術を効率よく吸収して発展してきました。戦後もこの傾向は基本的には変わっていません。現在、我が国は大量の自動車や電子機器を輸出して経済大国となっていますが、これらの工業製品の基本原理はほとんど外国で考えられたものです。欧米諸国との間に経済摩擦や文化摩擦が生じている現状を考えると、これからの我が国で大切なことは独創性のある個性的な人材を育成して、新しい科学技術のフロンティアを切り開き、世界に貢献することであると考えられます。

千葉大学では、日本のみならず、世界の科学技術の先端を担う若者を発掘し、育成するための一助として、本年度も、第28回数理科学コンクールを開催しました。

過去3回の経験から、課題の部は、遠隔開催も引き続き実施することとしました。そして、昨年度に引き続き、本年度も「人工知能の部」を開催しました。人工知能の部は完全に遠隔開催としました。

AI, データ・サイエンスに関する報道は落ち着きを見せてきています。AI, データ・サイエンスの手法がいろいろな現場で普通の考え方になってきました。また、AI, データ・サイエンスの課題を取り扱い、解決するためのプログラムも種々公開されています。そこで、千葉大学数理科学コンクールでは、令和5度より「人工知能の部」を開催しました。

数理科学コンクールの主題である、「自ら実験をして現象を考察する。」を引き続き実施するために、遠隔開催参加者には実験機材を送ることにしました。配送できる機材の大きさや種類には制限があるため、皆さんが受け取る機材の中には自分の使いたいものが入っていないかもしれません。しかし、与えられた機材だけを使って、実験の方法を考察して工夫することも科学者にとって重要な訓練です。今回から、対面開催の参加者と遠隔開催の参加者とに共通の4つの課題に取り組んでもらいました。対面開催への参加者は従来通りに、コンクール当日の提出締切時間までに、答案を提出してもらいました。また、遠隔開催への参加者は、開催案内に書かれている期日までに解答を千葉大学に返送してもらいました。

人工知能の部は、課題を登録したメールアドレスに送信し、解答をメールに添付して提出してもらいました。

このコンクールの特色は次の通りです。

1. 自由にゆったり考える

会場での試験時間は1日、自宅で時間を自由に使い、解答を導く。また、インターネットの検索も条件を付けて可としました。

2. たのしい物理・数学の発見

物理や数学のカリキュラムにとらわれず、物理や数学の本質に根ざした、考えて楽しい問題を提供する。

3. 多彩な才能の評価

様々な参加者の優秀な能力やユニークな発想を多面的に評価するため、問題をたくさん解いたものだけでなく、1題に集中してすばらしい発想を出したものも表彰の対象にする。

4. 人材の育成

コンクール参加者の物理や数学の能力をさらに高めるため、ホームページ上に表彰者の名前と講評を掲載する。

過去 27 回のコンクールに引き続き、多くの中高生の参加者があり、いろいろユニークなアイデアが生まれました。中学生も、高校生に負けず優秀でありました。そして、答案を見ると、それぞれの問題に興味を持ちながら解答していることが読んでとれました。

第 28 回数理科学コンクールの課題の解説と提出された答案の評価を以下にまとめます。解説に述べてあるように、各課題は課題出題者の周りにある基本的な問題や最先端の問題、さらには歴史的に意味のある問題を元にして作成しました。課題提出者一同、みなさんの素晴らしい洞察力と表現力を前にして、大変感心いたしました。

参加者の皆さんが今後、科学する心を磨き続け、我国の科学の発展に貢献することを課題作成者一同希望します。今後も諸君と共に科学することを楽しみたいと考えています。千葉大学では今後も引き続きこのコンクールを実施する予定です。物理・数学に興味がある中高生の積極的な参加を期待しています。課題作成者もさらに研鑽をかさね、おもしろく、しかも科学の本質に迫る課題を考えていきます。

課題作成者

千葉大学名誉教授 井宮 淳
東京慈恵会医科大学教授 植田 毅
(五十音順)

令和 7 年 11 月 2 日

2 優秀者氏名

令和7年8月1日に開催した第28回数理科学コンクール課題の部, 8月21日から27日の期間開催した人工知能の部, それぞれの参加者の皆さんのすばらしい答案の中から以下の参加者諸君を表彰することを決定しました.

第28回数理科学コンクール優秀者

課題の部

金樺賞	若杉怜奈		
	加藤結羽	佐々木 結	落合 諤恵
銀樺賞	大中洋奈	村川博美	
	白戸晴瑛	中田啓之祐	藤田稟生
	都築凜斗	本山松明	江口隼平
	目黒愛苺	柴田倅奈	佐藤瑠泉
	山本喜一	熊切優太	渡辺悠貴
	洲崎泰彰	安藤慧悟	黒岩隆悟
	伊佐菜々香	岩田ののか	
	田村和也	金田直人	河野一輝
学長賞	今枝駿斗		
	柴田心春		

人工知能の部

知巧賞	伊藤 尚太
-----	-------

課題	参加者氏名		
1	若杉怜奈 今枝駿斗 白戸晴瑛 都築凜斗 山本喜一 伊佐菜々香	中田啓之祐 本山松明 熊切優太 岩田ののか	藤田稟生 江口隼平 渡辺悠貴
2	若杉怜奈 柴田心春 大中洋奈 洲崎泰彰 田村和也	村川博美 安藤慧悟 金田直人	黒岩隆悟 河野一輝
3	今枝駿斗 加藤結羽 目黒愛苺	佐々木結 柴田倅奈	落合謠恵 佐藤瑠泉
4	加藤結羽	佐々木結	落合謠恵

事前に了承を得た方の氏名のみ掲載しております。

千葉大学先進科学センター	
センター長	教授 寺内文雄
千葉大学情報・データサイエンス学府	
千葉大学情報・データサイエンス学部	
学府長・学部長	教授 塩田茂雄
千葉大学大学院融合理工学府数学情報科学専攻	
専攻長	教授 川本一彦
千葉大学情報戦略機構	
機構長	教授 今泉貴史

課題の部

3 課題 1(対面・遠隔共通課題)

ガリレオ・ガリレイは、ピサの斜塔の上から質量の異なる2つの物体を同時に落とし、それらが同時に地面に到達することを観測したと伝えられています。では、斜面を滑り降りる物体の場合はどうでしょうか？床面が板でできている滑り台と、図1に示すローラーが並んでいる滑り台(ローラー式滑り台)では滑り降りる時間に違いはあるのでしょうか？物理的に考察してください。特に、体重が異なる人が滑った場合にどのような違いがあるのかについては考察しておいてください。

解説

滑り台は誰もが遊んだことがあるでしょう。なぜこの課題を出題した理由は子供を連れて森林公園へ行ったとき、200m くらいのローラー滑り台があり、滑って遊んだときに、子供は嬉々として遊んでいたのに、私自身はスピードが出過ぎて怖い思いをしたためです。明らかに体重が重い方がスピードが出ている。しかし、普通の滑り台ではそんなことにならないことは高校物理の範囲で理解できます。では、普通の滑り台とローラー滑り台との違いは何かを考えてもらうのが目的です。

この課題の実験は

村田次郎, 塩田将基, すべり台の動摩擦係数の実測研究, 物理教育, 第71巻, 第2号(2023)で詳しく調べられています。この論文では、ローラー式の滑り台については「重い人ほど速く滑る」ことを確認し、通常の滑り台ではそうではなく、「有意にクーロンの法則(高等学校で習う摩擦の法則)からずれる事を示す結果は得られなかった」としている。この論文は実験的研究で、なぜそうなるのか、理論的考察が足りません。このコンテストではなぜそうなるのかを考えて欲しかったのです。

まず、普通の滑り台について考えましょう。水平面からの角度が θ 、動摩擦係数 μ の斜面を滑る質量 M の直方体の箱を考えると摩擦力はクーロンの法則にしたがうと、箱が斜面から受ける垂直抗力 N は

$$N = Mg \cos \theta$$



図 1: ローラー式滑り台

であり, 動摩擦係数 μ は

$$f = \mu N = \mu Mg \cos \theta$$

となります.

斜面に平行な方向の運動方程式は

$$Ma = Mg \sin \theta - \mu Mg \cos \theta$$

$$a = g (\sin \theta - \mu \cos \theta)$$

(ここで, a は斜面に平行で下向きの箱の加速度, g は重力加速度) で与えられます. クーロンの法則に従う限り, 摩擦係数 μ も箱の質量 M にするため, 加速度は質量に依りません. 通常の滑り台での滑り方は体重に関係ないことになります.

もちろん, 体重が重い人は体が大きく, 空気抵抗が大きいなどの効果を考慮すれば少し変わってくるかもしれませんが. しかしそれは, 体重が重い人の方が遅くなるという結果になります.

さて, 次にローラー式の滑り台について考えてみましょう. この問題は高等学校の物理では扱わない (剛体の) 回転運動を考える必要があります. まず, 回転運動の一般論を簡単に説明しておきます. 出題者が高等学校の生徒時代には高校物理の範囲でした.

高等学校で習うニュートンの運動方程式は質点の加速度に質量をかけたものが加えた力になることを表しています. 実際には, 運動量 (質量と速度の積) の時間変化が力に等しいという式です.

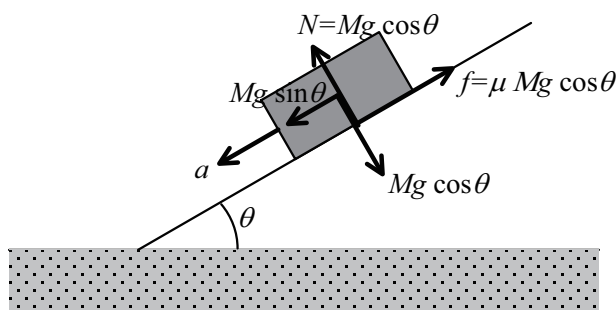


図 2: 普通の滑り台の斜面模型

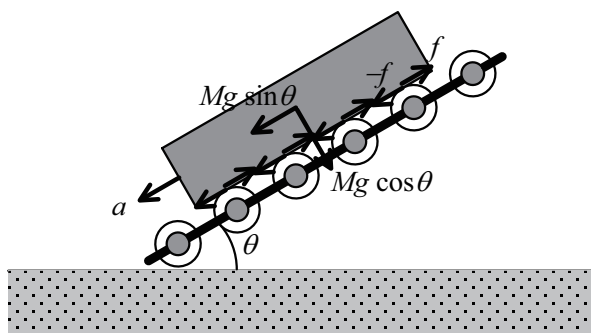


図 3: ローラー式滑り台の模型

他方、回転について、ビスを回すときにドライバーを用いますが、ドライバーのグリップが細いものよりも太いものの方が小さな力でビスを回すことができます。バレーボール、テニス、バドミントンのネットを張る時、ワイヤーをリールでハンドルを回して巻きますが、軸を直接持って回すことはできません。ハンドルも柄の長いものの方が小さな力で巻くことができます。これは、回転では、並進運動(質点のニュートンの運動方程式)のように力そのものではなく、力と力を加えている点(力点)と回転軸の距離の積によって決まることを表しています。この、物体(剛体)を回転させようとする「力 F と力点と回転軸の距離 r の積」を力のモーメント(トルク) N と呼びます。実際には、力、力のモーメントもベクトルで定義されます。力のモーメントの大きさは

$$N = Fr \sin \phi$$

で与えられます。ここで、 ϕ は力と力点から回転軸におろした垂線のなす角です。

単位時間(通常、1秒)あたりに回転する角度を角速度 ω 、単位時間あたりの角速度の変化を角加速度 ζ と呼びます。並進運動のニュートンの運動方程式に対応する回転の運動方程式は

$$I\zeta = N = Fr \sin \phi$$

と表されます。つまり、加速度が角加速度、力が力のモーメント、質量が I に置き換わったものになります。(実際には、角運動量(I と角速度の積)の時間変化が力のモーメントという式) 質量は運動の状態の変化のし難さ(静止している場合は動き難さ)を表しますが、 I は回転の状態の変化のし難さを表し、慣性モーメントと言います。慣性モーメントは、物体を細かく分割して、質量 m_i の i 番目の欠片が回転軸からの距離 r_i にあるとき

$$I = \sum_i m_i r_i^2$$

で与えられます。実際には和ではなく積分で表されます。慣性モーメントは同じ物体でも回転軸が異なれば値は異なります。

質量 m の小さな球に長さ l の軽い糸を付けて、糸の他端を回転の中心に固定して回すときの慣性モーメントは

$$I = ml^2$$

となります。半径 r 、質量 m の一様な円柱の円柱の軸を回転軸として回転させるときの慣性モーメントは

$$I = \frac{1}{2}mr^2$$

となります。

さて、ローラー式滑り台の問題を考えましょう。箱に同時に n 個のローラーが接触しているとします。ローラーを質量 m の一様な円柱とします。話を簡単にするため、ローラーと軸との摩擦は考えないことにします。

箱が1つのローラーの表面に力 f を及ぼすとする、箱は逆向きのその反作用 $-f$ の力を受けます。箱は n 個のローラーと接しているので、全体で $-nf$ の力を受けます。箱の運動方程式は

$$Ma = Mg \sin \theta - nf$$

ローラーの回転の方程式は

$$I\zeta = fr$$

となります.

また, ローラーが箱に接して滑ることなく回転するすると, 角度 (ラジアン) の定義から, 周長 = 半径 $\times$ 内角, 速度 = 半径 $\times$ 角速度, $a = r\zeta$ となる. この式を用いて, 2つの運動方程式から f, ζ を消去すると

$$\begin{aligned} Ma &= Mg \sin \theta - n \frac{I}{r} \zeta \\ &= Mg \sin \theta - n \frac{I}{r^2} a \\ \left(M + n \frac{I}{r^2} \right) a &= Mg \sin \theta \\ a &= \frac{M}{M + n \frac{I}{r^2}} g \sin \theta \\ &= \frac{M}{M + n \frac{\frac{1}{2}mr^2}{r^2}} g \sin \theta \\ &= \frac{1}{1 + n \frac{m}{2M}} g \sin \theta \end{aligned}$$

となります. この場合, 箱が滑り降りる加速度は箱の質量 M に依存することが分かります. この式は箱の質量が大きい方が加速度は大きく, 滑る人の体重が重い方が滑る加速が大きくなることを意味しています.

普通の滑り台の加速度は $a = g(\sin \theta - \mu \cos \theta)$ であるので, 以下の条件が成り立つとき, ローラー式の滑り台の方が速く滑ることになります.

$$\begin{aligned} \frac{1}{1 + n \frac{m}{2M}} g \sin \theta &> g(\sin \theta - \mu \cos \theta) \\ \mu \cot \theta &> \frac{\frac{nm}{M}}{2 + \frac{nm}{M}} \end{aligned}$$

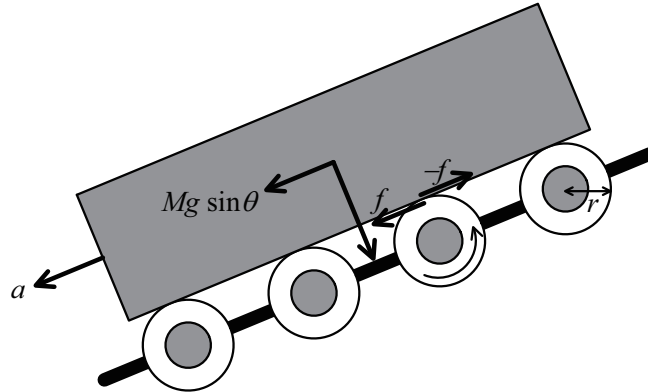


図 4: ローラー式滑り台の模型 (拡大図)

滑る人の体重 M は体に乗っている全てのローラーの質量 nm よりもずっと重いと考えられ、 $nm < M$ の場合、 $\frac{nm}{2 + \frac{nm}{M}} < 1/3$ となります。

また、斜面の角度 $\theta < 45^\circ$ の場合、 $\cot \theta > 1$ であり、軟鋼と木綿の乾燥摩擦係数 μ が 0.40 であるので、この条件はほぼ常に成り立っていると考えられます。したがって、ローラー式の滑り台の方が速く滑ると言えます。

ここでは、大学教養レベルの簡単な取り扱いをしましたが、車輪 (ローラー) が転がる運動は複雑でいろいろ考えなければならぬことが多い問題です。レールやすべる物体の変形によるローラーとの接触面積の変化、ローラーの変形、ローラーと以下の論文によくまとまっていると思います。参考にしてください。

1. 佐々木祥介, 南波陽平, 岩成唯夫, 北野保行:円柱の転がり運動, 大学の物理教育,

https://ir.library.osaka-u.ac.jp/repo/ouka/all/88664/peu_27_1_26.pdf :無料でダウンロード可能

2. 江沢 洋, 上條隆志 編:江沢 洋 選集 第1巻 物理の見方・考え方 (日本評論社)11 章.

講評

解答にネット検索で物理教育の論文を見つけそれをまとめたものがありましたが、国語の試験ではないのでただまとめただけでは評価できません。そこから自分たちでどう考え、納得できる理由を示して欲しかったですね。

実験用にローラーレール2台とローラーレールの代わりに算盤を用意してありました。算盤をその目的で利用した人はそれほど多くなかったようですが...

ローラーレールを使って、計画的な実験を行った解答がありました。結果、結論はともかく、推論と実験計画を高く評価しました。

4 課題 2(対面・遠隔共通課題)

温度を測る道具としては、小学校の理科の実験でも用いる灯油の熱膨張を利用した棒温度計がなじみ深いかもしれません。この温度計を図5左に示します。新型コロナの流行以降、体温計も赤外線を用いた非接触型の放射温度計(サーモグラフィー)が普及しました。放射温度計を図5中に示します。また、透明な液体中に浮力の異なる複数の浮き子を入れ、液体の密度の温度変化により浮沈する浮き子に記された温度で大まかに温度を知るガリレオ温度計もインテリアとして販売されています。ガリレオ温度計を図5右に示します。液晶で温度に応じた色や数字が浮き出るサーモテープなどもあります。科学実験や工業製品では、放射温度計、抵抗温度計、熱電対、バイメタル式温度計など場合に依じて様々な測定原理の温度計が用いられています。

液体について、温度計を用いず、液体に直接接触らず、これまでにない原理で温度を測る方法を考えてください。液体を移動させる(流す)ことは構わないことにします。

解答にはどのような物理法則を用いて、どのような原理で測定できるのかを明確に記してください。理科年表などの物性値を用いて測定精度などを考察していれば高く評価します。



図 5: 棒温度計 (モノタロウ), 放射温度計, ガリレオ温度計 (WikiPedia)

解説

問題文にもあるように、温度計は測定対象物もしくは測定に用いる物質の物性(物の性質)の温度変化を利用しています。新しい原理の温度計を考える場合、どんな物性を用いるか？物質が十分な温度変化を示すか？この問題の場合、流体と言ってるので、流体の特徴的な物性としては、密度、表面張力、粘性が考えられます。表面張力、粘性は多くの液体で温度が変わると、測定できる程度に変化するので利用できそうです。この中で接触せずに測定できるものは何かを考えなければなりません。どのようにしたら、接触せずに測定できるのか？測定方法も考える必要があります。

静止している流体であれば、容器の端のメニスカスは表面張力に依るので温度によって変化するので大きさ、形を見れば温度が分かる可能性があります。また、容器に入れた液体の表面に音波を当てて、波を発生させれば、液体の表面に現れる波の波長は表面張力に依存するでしょう。表面に立つ波の波長、波の模様は温度により変化する可能性があります。

問題文には液体を流しても構わないとわざわざ断っているのですが、流れている液体を考えると、液体を流す圧力が同じ場合、流速は粘性に依存します。水を流すのと蜂蜜を流すのだと速さが違うのと同じです。また、冷蔵庫から出したばかりの蜂蜜はなかなか流れませんが、出してから時間が経ち、室温になると流れやすくなります。粘性が大きく温度変化する例です。半径 a 、長さ l の管の両端の高さの差が H で圧力差 ΔP がかかっているときに、その管を密度 ρ 、粘性係数(粘性率) η の流体が流れるとき、1秒間に流れる流量(流体の体積) Q は

$$Q = \frac{\pi a^4}{8\eta l} (\Delta P + \rho g H)$$

で与えられます。 g は重力加速度です。これから、管に流体を流せば流れる量から温度が分かります。

この課題を出題は以下の論文に端を発します。

Mohr Wenger, Amber Maimon, Or Yizhar, Adi Snir, Yonatan Sasson and Amir Amedi:
“Hearing temperatures: employing machine learning for elucidating the cross-modal perception of thermal properties through audition,” *Frontiers in Psychology*, Vol. 15
(2024) DOI:10.3389/fpsyg.2024.1353490

無料でダウンロードできます。英語ですが Google 翻訳でも ChatGPT でも翻訳してくれますから読んでみてください。

数年前にこの数理科学コンクールで水琴窟を題材とした課題を出題しましたが、溜まった液体に液体を垂らすと音が発生します。発生する音は粘性により変化します。温度が高い場合の音と温度が低い場合の音を機械学習させ、ある状態での音をどちらの音か判断させると区別がつくという研究です。

これを細かく行うことで、液体が滴る音だけから温度を測定できるであろうということで出題しました。

講評

利用する物性としては粘性を用いようとした解答が多くみられました。温度が変化すると十分粘性が変化することを確認した解答もありました。これが確認できていないと机上の空論になりますから、重要なことです。それをどう利用するかはいろいろありましたが、先に挙げた粘性がある場合の流量の式を用いて温度が分かるという解答がありました。流量をどう測定するかによって接触か非接触化が変わってきますが、用いる式を明示していたことを高く評価しました。

5 課題 3(対面・遠隔共通課題)

現在 NHK で放送中の連続テレビ小説「あんぱん」では”図案”と呼ばれる美術の分野が出てきます。狭い意味での図案は商業絵画であり、ポスター、ラベル、包装紙、などの意匠 (デザイン) を指します。公共空間での禁煙が今ほど厳しくない時代には喫茶店で店の広告を施した小さな燐寸箱が配られていました。これらは、街の小さな印刷会社で木版や石版で印刷されていました。商品の包装を少し広く捉えると、商品を納める箱、店の紙袋のデザインも図案の分野に入ります。これは package design と呼ばれます。さらに、箱のデザインは、商品の収納法のデザインも含みます。例えば、箱を開けたときに中に詰められた商品の見栄えが良い詰め方も考えます。

磁器製品、陶器製品、ガラス製品等、運搬中に破損しやすい物品の場合には、その収納法、緩衝材の入れ方を考える必要があります。緩衝材として長く、発泡スチロールが利用されてきました。それ以前は、粃殻や木毛 (木を薄く削った糸状材) が利用されてきました。現在では、資源保護の観点から、再生可能な、紙、段ボールだけで、衝撃材と一体型の梱包が一般的になってきました。このような箱では、箱加工の観点から考えると、切込や、折込が最小限であることが必要です。また、梱包作業時に折込が容易に行える必要があります。

梱包を考える上で、基本となるものの 1 つがカトラリー (洋食器) です。カトラリーの 1 人用基本セットは、大スプーン、大フォークと対のナイフ、ティースプーンの 4 種類です。セットの場合は、5 人分または 6 人分で販売されています。また、美術品の輸送では、埴輪や仏像のような破損しやすい物品を梱包する必要があります。

問 1. 4 種のカトラリー 1 組を固定して、見栄えよく、1 枚の紙に、切込を入れ、最小限の折込で、固定して収納できる、衝撃材と一体型の箱を考えなさい。

問 2. 4 種のカトラリー 6 組を固定して、見栄えよく、1 枚の紙に、切込を入れ、最小限の折込で、固定して収納できる、衝撃材と一体型の箱を考えなさい。

問 3. 埴輪のような破損しやすい物品を固定して運送する為に 1 枚の紙に、切込を入れ、最小限の折込で、固定して収納できる、衝撃材と一体型の箱を考えなさい。

解答は、折込の山線、谷線と、切込線を定義し、図面で提出してください。

解説

カトラリーの配置を数学的に考察してみよう。4 本組のカトラリーの配置を関数で表すと

$$f(n) = \begin{cases} k & n = 4m + 1 \\ f & n = 4m + 2 \\ s & n = 4m + 3 \\ t & n = 4m \end{cases} \quad (1)$$

と考えることができる。ここで、 k, f, s, t は knife fork spoon tee-spoon の頭文字です。

この対応関係を横に並べると

$$\begin{array}{cccccccccccccccc} \mathbf{S} & = & m & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \cdots & 4(n-1)+1 & 4(n-1)+2 & 4(n-1)+3 & 4n \\ & & f(m) & k & f & s & t & k & f & s & t & \cdots & k & f & s & t \end{array}$$

$$= (kfst)^n \quad (2)$$

となります. ここで, $(kfst)^n$ は 4 本組を n 組繰返し並べることを意味します. 式 (1) は周期 4 の周期関数であることが分かります.

この周期関数をどのようなデータ構造で表現するかを考えてみます. 同じものだけを積重ねて配置すると

$$\mathbf{H} = \begin{matrix} 1 & k & f & s & t \\ 2 & k & f & s & t \\ \vdots & & & & \\ n & k & f & s & t \end{matrix} = \begin{matrix} k^n & f^n & s^n & t^n \end{matrix} \quad (3)$$

と成ります, さらに, これを円環状に配置すると

$$\mathbf{C} = \begin{pmatrix} k & & \\ t & f & \\ & s & \end{pmatrix}^n = \begin{pmatrix} k^n & & \\ t^n & f^n & \\ & s^n & \end{pmatrix} \quad (4)$$

と成ります.

夫々の配置に従って, 固定法を考えることになります. 販売や輸送のための固定法だけでなく, デザイン (意匠) 性も含め, 付加価値を付けたカトラリーセットを考案することが現在行われています.

新潟県燕市のメーカーが提案する梱包も含めたカトラリーが以下の url から見ることができます.

<https://www.sunao-tsubame.com/shopdetail/000000000199/007/X/page1/recommend/>

以下の url に上の数学的解析を反映したような高級カトラリーセットが掲載されています.

https://www.christofle.com/ja_jp/concorde

https://www.christofle.com/ja_jp/couverts/collections/mood.html

壊れ物を輸送のために梱包する基本的であり, しかも広く使われている方法は梱包の中に壊れ物を浮かせることです. これは, 振動を吸収するためです. 精密機械を運搬する装置として無振動車があります.

課題は, このような条件を満たす梱包を考えることです. 通常, 梱包箱は 1 枚の紙 (段ボール) を折り込んで作成されます. 折込んだ部分をどのように利用して梱包物を浮かすかを工夫します. 1 枚の紙 (段ボール) から立体的な箱を作るためには, 折り込んで, 底や壁, 蓋の部分を折りだす必要があります. この部分を利用して梱包物を浮かすことを考えます. この問題は, 折紙工学の問題となります.

講評

カトラリーの梱包に関しては従来の梱包にない発想を図面で表現した解答を評価しました. また, 植輪の梱包に関しては, 梱包箱を作成する場合の余白を利用して梱包物を浮かせるかを考えた解答を評価しました.

6 課題 4(対面・遠隔共通課題)

第1回数理科学コンクールに、次に述べる課題を出題しました。

各頂点が自由に動く立方体の骨組を考えます。この骨組の頂点間に筋交いを補強する場合、筋交いの総延長が最小になる補強の方法を考察しなさい。

骨組の部材は長さと質量があり、太さの無い理想的な柱である場合には、頂点の位置ベクトルと筋交いの長さから決まる行列の性質を解析すれば答えがでます。

今回は、実際の立方体を作成してこの課題に取り組みます。各頂点では、最初3本の柱が交わっています。筋交いを追加すると、4本の柱が交わります。実験をしてこの課題の解を求めるためには、複数の柱が頂点で自由に動く接続構造を作る必要があります。そこで、太さのある複数の柱が接合点で自在に動く接合法を考えてください。ここで、“自在に動く”とは一方の柱の動きを固定した場合に、固定していない柱が空間的に 4π の立体角を動くことです。

遠隔参加で、解答として写真や動画を提出する場合には、

subject (件名) を”課題4 解答 (参加者番号)(氏名)”として

cfs-contest@chiba-u.jp

宛て、フィル名を AI(参加者番号).(拡張子) にして送信してください。

また、解答として実物の提出を希望する場合には別途自己負担で返信してください。

解説

頂点と頂点間を結ぶ辺からなる図形をグラフと呼びます。頂点がユークリッド空間の座標であるグラフを幾何グラフと呼びます。幾何グラフの典型的なものは理想的建築構造体です。理想的建築構造体とは筋交いなどが無限に細い辺であり、しかも辺の重さが無視できるものです。このことから建築構造体の性質を幾何グラフの数学的性質を利用して解析できることが分かります。このとき、頂点では辺が自由に稼働できることを仮定しています。

特に重要な幾何グラフの性質は幾何グラフの剛性です。幾何グラフの剛性とは各頂点の微小移動に関して、幾何グラフが安定なことです。すなわち、幾何グラフが動かないことです。

2次元ではこの問題は解決していますが、3次元以上では未解決な問題がたくさんあります。3次元構造体として重要な対象は、遺伝子、タンパク質、鎖状高分子などです。これらの研究では剛体マトロイド理論を使います。マトロイドとは行列の列ベクトルの概念を一般化し、集合の包含関係、順序関係をもとに、要素の独立性を解析する数学の1分野です。その理論を幾何グラフに適用して安定性を解析します。

3次元の簡単な構造体に対しては、実験によって解決することができます。ただし、このとき頂点に、辺が自由に動く接合装置を用意する必要があります。逆に、接合点が自由に動く必要のある幾何グラフとして、クランク機構やロボットアームがあります。

講評

自由継手を図面で分かりやすく説明している解答を評価しました.

人工知能の部

7 人工知能の部 課題

生物の種の同定と分類のためにゲノムプロファイリング (GP) が用いられます。GP は, (1) ランダム PCR, (2) マイクロ温度勾配ゲル電気泳動 (μ TGGE), そのゲノムプロファイルの特徴点の推定と処理によって, (3) 種同定ドット (スピド) を生成する, という 3つの主要ステップから構成されます。

μ TGGE は, ポリアクリルアミドゲル中で温度勾配を形成し, 生体高分子を泳動させることで, 分子の構造変化や変性を利用して分離を行います。温度上昇に伴い, 高分子は変性し, その形態が変化します。そのため電気泳動における移動度に違いが生じ, 分離が可能になります。

μ TGGE では図 1 のような画像が得られます。縦軸方向に電位勾配をかけ, 縦軸が移動度, 横軸が温度です。

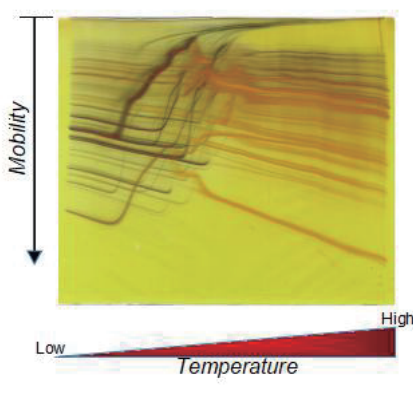


図 6: マイクロ温度勾配ゲル電気泳動画像 (DOI:10.26434/chemrxiv-2024-n9q36)

μ TGGE では, 電気泳動バンド (黒い線) の位置が温度によって変化し, DNA 断片の固有の融解温度を超えると, DNA の変性によりバンドの軌跡が急激に変化します。急激に変化する点をスピドと呼ぶます。全てのバンドのスピドの位置を正確に推定することが重要になります。処理すべき画像は大量にあるので, この作業を自動化することを考えます。

電気泳動画像に現れる全てのバンドのスピドの位置を正確に推定することが重要になります。処理すべき電気泳動画像は大量にあるので, Mathematica の機械学習機能を用いて, できるだけ多くの推定用データのスピドの座標を求めてください。いずれのデータの電気泳動画像にはスピドを緑と黄色の $\circ$ で表示しています。スピドの座標を求める場合に, これらの電気泳動画像から直接 $\circ$ 図形の座標を求めるアルゴリズムを用いないでください。

問 1 Mathematica の機械学習機能を用いて, 推定用データの電気泳動画像のみから, そのスピドの座標をできるだけ多く求めてください。計算に使用した notebook を.nb 形式のまま提出してください。

問 2 電気泳動画像から, スピドの座標をどのように求めたかアルゴリズムの考え, 動作等を説明するレポートを纏めてください。

問 3 問 1 で求めたスピドの座標と推定用データとして与えたスピドの座標を比較することにより, 推定精度の検証を行ってください。推定精度の検証計算に使用した notebook を提出してください。

1. 学習用データと推定用のデータは Mathematica notebook の形で Wolfram Cloud 上で共有します.
2. 学習用データには 25 枚の電気泳動画像とそれらの全てのスピドの座標がリスト形式で収められています.
3. 推定用データには 5 枚の電気泳動画像とそれらの全てのスピドの座標がリスト形式で収められています.
4. 学習用データ, 推定用データ, いずれの電気泳動画像にも便宜的にスピドの位置を緑もしくは黄色の $\circ$ で表示していいいます. 実際の電気泳動画像には緑もしくは黄色の $\circ$ は付されていないことを念頭に解答してください.

本課題で使用するデータは 西垣功一 埼玉大学名誉教授にご提供頂きました.

解説

この課題は問題文にもある通り, 今現在進行形の研究で必要とされている技術です.

1. Shamim Ahmed, Manabu Komori, Sachika Tsuji-Ueno, Miho Suzuki, Akinori Kosaku, Kiyoshi Miyamoto, Koichi Nishigaki: (2011) Genome Profiling (GP) Method Based Classification of Insects: Congruence with That of Classical Phenotype-Based One. PLOS ONE 6(8): e23963.
<https://doi.org/10.1371/journal.pone.0023963>
2. Shamim Ahmed, Manabu Komori, Sachika Tsuji-Ueno, Miho Suzuki, Akinori Kosaku, Kiyoshi Miyamoto, Koichi Nishigaki:(2025) Method and theory of genome profiling (GP) developed for identification and classification of organisms, 14 May 2025, This content is a preprint and has not been peer-reviewed.
<https://chemrxiv.org/engage/chemrxiv/article-details/67ac1c54fa469535b9f7b556>
<https://doi.org/10.26434/chemrxiv-2024-n9q36-v3>

話の発端は 2024 年 3 月に関西学院大学で開催された第 96 回形の科学シンポジウムで西垣功一 埼玉大学名誉教授からスピドを Mathematica の機械学習機能で簡単にスピドの座標を求めることはできないかと相談されたことに始まります.

ネットで検索してみると,

鈴木雅人 阿部巧 レウエティ要, 北越大輔: 適応的二値化処理に基づく温度勾配ゲル電気泳動画像のスピド自動検出法, 情報処理学会 第 87 回全国大会講演論文集 (2025) 3 月

<https://www.ipsj.or.jp/event/taikai/87/WEB/data/pdf/7D-04.html>

という講演が見つかります。別の研究で問題を解決しようとしているのかと、西垣先生に知らせたところ、共同研究者だということで、他の方法で座標を求める試みもなされていますが、まだ、精度には問題があるようで、情報学的にもチャレンジングな問題となっています。

この話を聞いたときに、スピドの個数はともかく、密集して存在しているものでなければ数個の座標の特定は簡単に行えるのではないかと直感していたのでとりあえず、似たようなダミー画像を用いて試してみました。

関数のパラメーターに乱数を用いて図7のような温度勾配ゲル電気泳動の模擬画像を300発生させました。関数から黒いバンドの縦軸の極小点の座標が与えられるので、この画像にその座標をタグ付けて学習させます。

Mathematicaには数値を推定する関数Predictがあるが、この関数はスカラー(1つの数値)しか推定できないので、そのままでは2次元の座標を推定することはできません。そこで、工夫が必要になるが、もっとも単純には画像と横軸の座標、画像と縦軸の座標をそれぞれ分けて学習させ、横軸、縦軸の座標を別々に推定させます。この場合、同じ画像を2度学習させることになり非常に効率良くありません。また、縦軸と横軸の値に相関がある場合、その相関を見落とすことになり、位置の推定には不利になります。この方法でのノートブックを末尾のnotebook1に示します。その精度をランダムに発生させた10のテスト画像で検証しました。図8には正解を紫の点で、推定した点を緑で示してあります。これだけ単純な方法でも十分一致した推定ができていると思えます。

この問題の場合、複数のバンドがあるので、バンドが2つある場合を考えましょう。この場合のノー

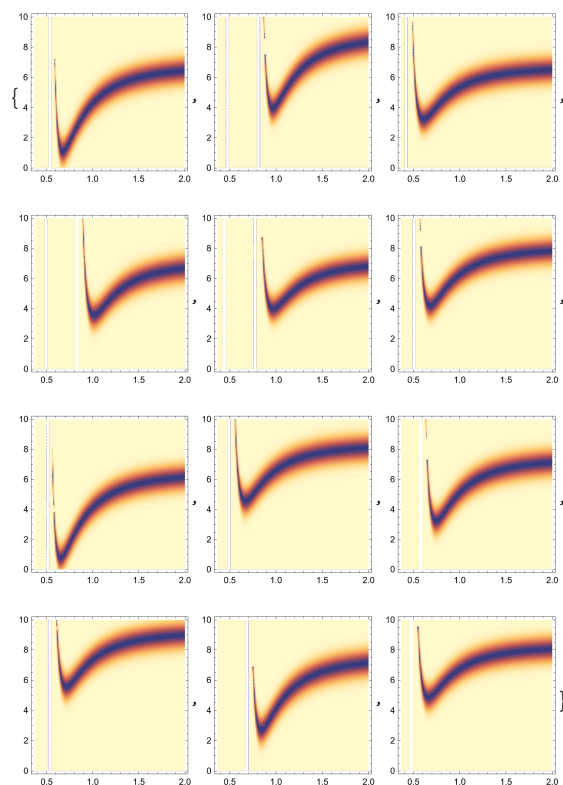


図7: 教師データ用模擬温度勾配ゲル電気泳動画像

トブックを末尾の notebook2 に示します。教師データ用画像は図 9 のようになります。このノートブックでは 2 つの極小点の座標の横、縦軸の座標を画像と関連付けて、それぞれ学習させているので、4 回同じ画像を学習させることになります。推定には、2 つの極小点の位置をそれぞれ別の学習データを用いて推定させています。推定結果は図 10 のようになります。推定結果はずれが目立つものもありますが、単純な方法にしてはよく推定で来ているように思えます。

実際のデータには多数のスピードが含まれるので、タグ付けする座標データを 2 つずつに分けて推定するようにすると、この方式で推定可能ではないかということで、試してみましょう。

また、教師データ画像の学習の順により結果が異なるので、学習の順序をランダムに変えたものを 25 通り用意し、それぞれデータを学習させ、推定した位置を示し、その平均位置を求めます。これを実行するノートブックを末尾の notebook3 に示します。5 つのテストデータの下から 2 つのスピードの位置を推定した結果を図 11 に示します。テストデータとして与えられたスピードの座標を紫、40 通りの順序で学習させて推定した最も下のスピードの位置を緑、下から 2 番目のスピードの位置をオレンジの点で示し、それらの平均をそれぞれ赤、青の点で示しています。学習順序により推定位置がかなりぶれることが分かりますが、平均位置は比較的正解をよく推定できていると思えます。

では、移動度が大きくなり、スピードが密集してくるとどうなるのかを見るために、下から 5、6 番目のスピードを推定させよう。これを実行するノートブックを末尾の notebook4 に示します。5 つのテストデータの下から 5、6 番目のスピードの位置を推定した結果を図 12 に示します。5、6 番目のスピードの

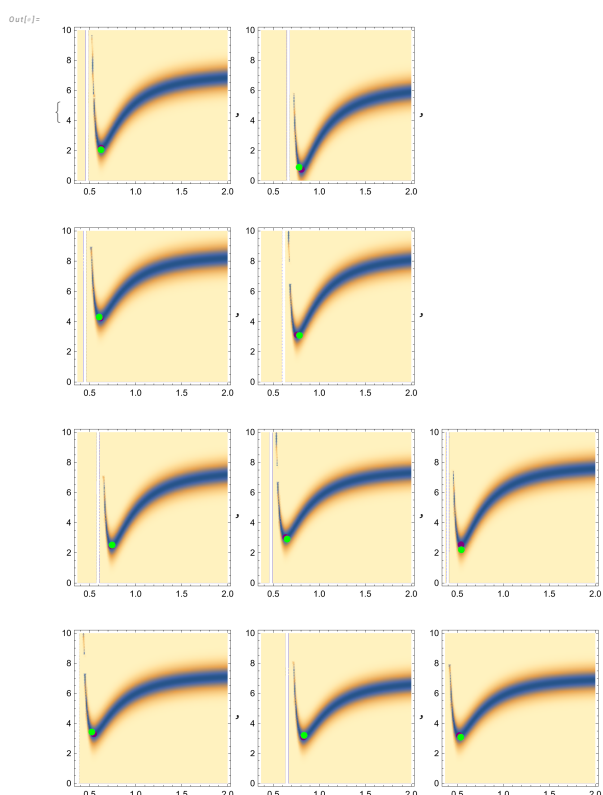


図 8: 温度勾配ゲル電気泳動模擬画像テストデータ。極小点の正解を紫、推定した位置を緑の点で示してある。

うちどちらか一方はよく推定できているが、もう一方は大きくずれている。どのテストデータも5, 6番目のスピドの縦位置が近いめ、分解が困難だったのかもしれませんが、やはり、スピドが密集している部分での位置特定は少なくともこの方法では難しいと言わざるを得ません。

これまで、Predict 関数を用いて、横、縦の座標を独立したものとして、学習させていましたが、この方法では効率が悪いうえ、横、縦の座標の相関が考慮されないため推定精度も悪いと考えられる。

そこで、画像に全てのスピドの座標をタグ付けしたものを学習させられるニューラルネットを自ら設定しよう。それを実行するためのノートブックを末尾の notebook5 に示します。

画像に下から6番目までのスピドの座標をタグ付けしたものを教師データとして入力する。第1層は10個のニューロンの線形層、5個の第2層、8個の第3層、座標の数値を6個出力する層からなるニューラルネットワークを構成します。この場合、学習は1度で済みます。定義したニューラルネットワークでスピドの座標を推定した結果を図13に示します。テストデータとして与えられたスピドの座標をオレンジ、推定されたスピドの座標をダークブルーの点で示しています。どの場合も6個のスピドの内いくつかは近い座標を推定しているけれど、いくつかは遠く離れています。また、推定されるスピドの位置は横軸が0.5付近に集まる傾向にあるように思われます。

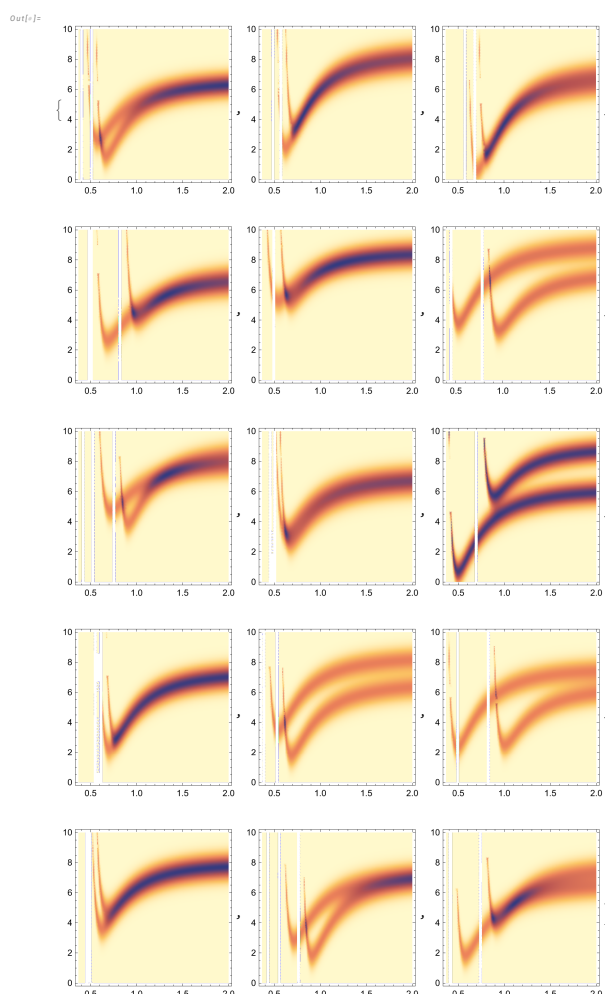


図 9: 教師データ用温度勾配ゲル電気泳動模擬画像 (バンドが2本ある場合)

この結果は, 10 個, 20 個, 出力層の場合でもほぼ同様であり変わりません. ネットワークの構成を工夫するとより良い推定ができる可能性もあります.

Wolfram Community のメンバーになれば, うまくいかないことを記事を投稿して相談することもできます. 興味を持ってもらえれば, 世界の人がいちろいんな解決策を提案してくれます. 場合によっては, Wolfram Research 社の技術者も返信してくれます. この問題も以下のように投稿してみました.

https://community.wolfram.com/groups/-/m/t/3332178?p_p_auth=saX6iCm1

今回については, コメントは全て既に試みてダメだったものでしたが....

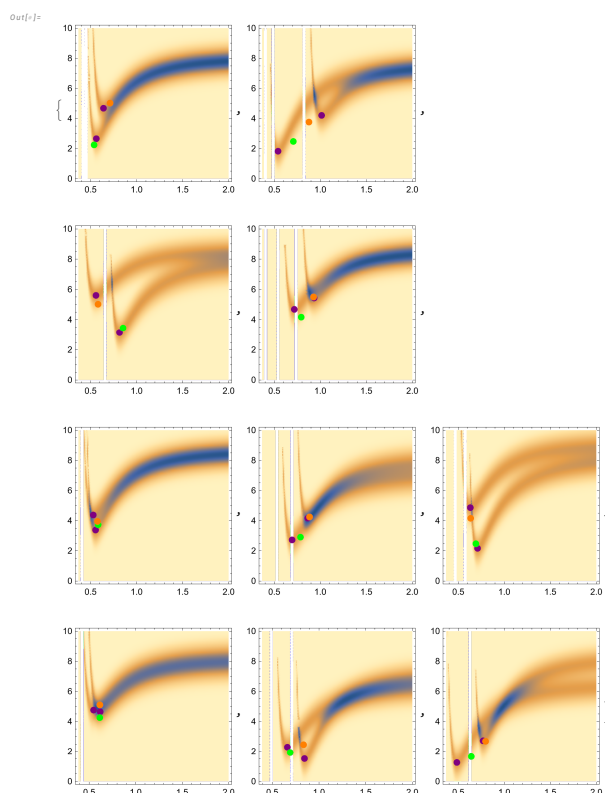


図 10: 温度勾配ゲル電気泳動模擬画像テストデータ (バンドが 2 本ある場合). 極小点の正解を紫, 推定した位置を緑とオレンジの点で示してある.

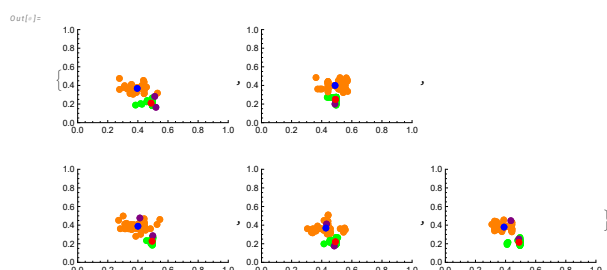


図 11: 温度勾配ゲル電気泳動画像テストデータのスピードの位置を推定した結果. テストデータとして与えられたスピードの座標を紫, 40 通りの順序で学習させて推定した最も下および下から 2 番目のスピードの位置をそれぞれ緑とオレンジの点で示し, それらの平均をそれぞれ赤, 青の点で示している.

実行したいことを数学的内容も含め ChatGPT, Gemini に聞けば, 実行手順を教えてくれ. Mathematica のコードも示してくれるので積極的に利用してみてください. 非常に有効なツールですが, 示された Mathematica のコードには存在しない関数が含まれたりすることがあるので注意してください.

講評

Mathematica の利点はそれほど長いプログラムを書かなくとも必要なことが実行できることにありますが, 提出された解答は Python など, 所謂プログラミング言語のように長々とプログラムを書いている例が多かったように思います. それほどプログラミングをすることなく目的は達成されるはずです.

このコンテストでは, プログラミングスキルの評価ではなく, 如何にスマートに, 正確に結果を出すかを評価しています. Mathematica の機能をフル活用してできそうにないことを簡単に実現してみようというのが狙いです. そういう意味で, それほどプログラミングスキルがない中学生でも Mathematica をうまく使えば太刀打ちできるだろうということを期待しています.

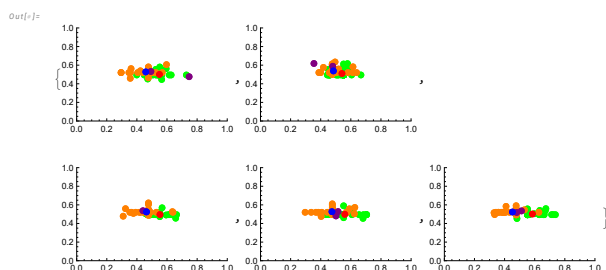


図 12: 温度勾配ゲル電気泳動画像テストデータのスピドの位置を推定した結果. テストデータとして与えられたスピドの座標を紫, 40 通りの順序で学習させて推定した下から 5, 6 番目のスピドの位置をそれぞれ緑とオレンジの点で示し, それらの平均をそれぞれ赤, 青の点で示している.

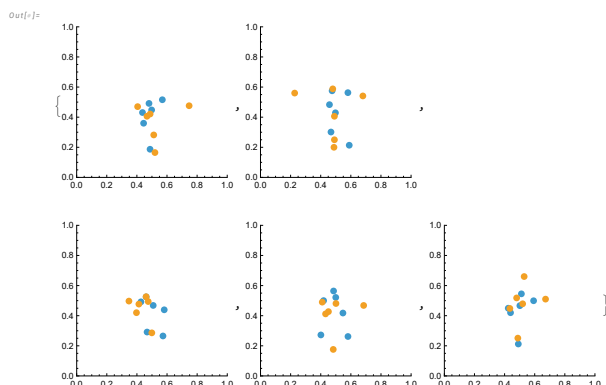


図 13: 温度勾配ゲル電気泳動画像テストデータのスピドの位置を定義したニューラルネットワークで推定した結果. テストデータとして与えられたスピドの座標をオレンジ, 推定されたスピドの座標をダークブルーの点で示している.

今回の課題は難しく考え込んだ気配があります。データのクリーニング、整形などあれこれ、データの前処理に時間を書けている例がありましたが、そんなことは気にせず、Mathematica の機能を信じでとりあえず、やらせてみれば結果はついてきます。やってダメならばその時点で考えればいいだけのこと。

精度はともかく、スピドの推定結果を出して欲しかったですね。

非常に精度よく推定できていれば、その結果は学会発表可能なのですから。

notebook1

```
In[*]:= nsp = 300;
        npr = 10;

In[*]:= rn = Table[{RandomReal[], RandomReal[], RandomReal[], RandomReal[]}, {i, 1, nsp + npr}];

In[*]:= si = Table[DensityPlot[
    1 - Exp[-(1.5 + rn[[i, 1]] 0.3) (y - (8 / 3 (0.4 + 0.2 rn[[i, 2]] / (v - 1 / 3 - rn[[i, 3]] / 2) -
    3 / (v - rn[[i, 3]] / 2) ^2 + 6 + 3 rn[[i, 4]])) ^2],
    {v, 1 / 3 + 1 / 30, 2}, {y, 0, 10}, PlotRange -> All, PlotPoints -> 50], {i, 1, nsp}];

In[*]:= pri = Table[DensityPlot[
    1 - Exp[-(1.5 + rn[[i, 1]] 0.3) (y - (8 / 3 (0.4 + 0.2 rn[[i, 2]] / (v - 1 / 3 - rn[[i, 3]] / 2) -
    3 / (v - rn[[i, 3]] / 2) ^2 + 6 + 3 rn[[i, 4]])) ^2], {v, 1 / 3 + 1 / 30, 2},
    {y, 0, 10}, PlotRange -> All, PlotPoints -> 50], {i, nsp + 1, nsp + npr}];

In[*]:= cpt = Table[sol = Solve[(0.4 + 0.2 rn[[i, 2]]) ==  $\frac{(-1 + 3 v)^2}{4 v^3}$ , v];
    {v + rn[[i, 3]] / 2,  $\frac{-2 + 3 v}{v^3} + 6 + 3 rn[[i, 4]]$ } /. sol[[2]], {i, 1, nsp}];

In[*]:= preas = Table[sol = Solve[(0.4 + 0.2 rn[[i, 2]]) ==  $\frac{(-1 + 3 v)^2}{4 v^3}$ , v];
    {v + rn[[i, 3]] / 2,  $\frac{-2 + 3 v}{v^3} + 6 + 3 rn[[i, 4]]$ } /. sol[[2]], {i, nsp + 1, nsp + npr}];

In[*]:= datasetx = Table[si[[i]] -> cpt[[i, 1]], {i, 1, nsp}];
In[*]:= datasety = Table[si[[i]] -> cpt[[i, 2]], {i, 1, nsp}];
In[*]:= px = Predict[datasetx, Method -> "GradientBoostedTrees"]
Out[*]=
PredictorFunction[ Input type: Image
Method: GradientBoostedTrees
Data not saved. Save now 
```

```
In[*]:= py = Predict[datasety, Method -> "GradientBoostedTrees"]
Out[*]=
PredictorFunction[ Input type: Image
Method: GradientBoostedTrees
Data not saved. Save now 
```

```
In[*]:= corpre = Table[{px[pri[[i]]], py[pri[[i]]]}, {i, 1, Length[pri]}]
Out[*]=
{{0.624223, 2.0541}, {0.78035, 0.89392}, {0.604401, 4.30797},
{0.785439, 3.07853}, {0.744666, 2.52008}, {0.65028, 2.91006}, {0.544197, 2.20075},
{0.523838, 3.43468}, {0.834665, 3.21342}, {0.537383, 3.0896}}

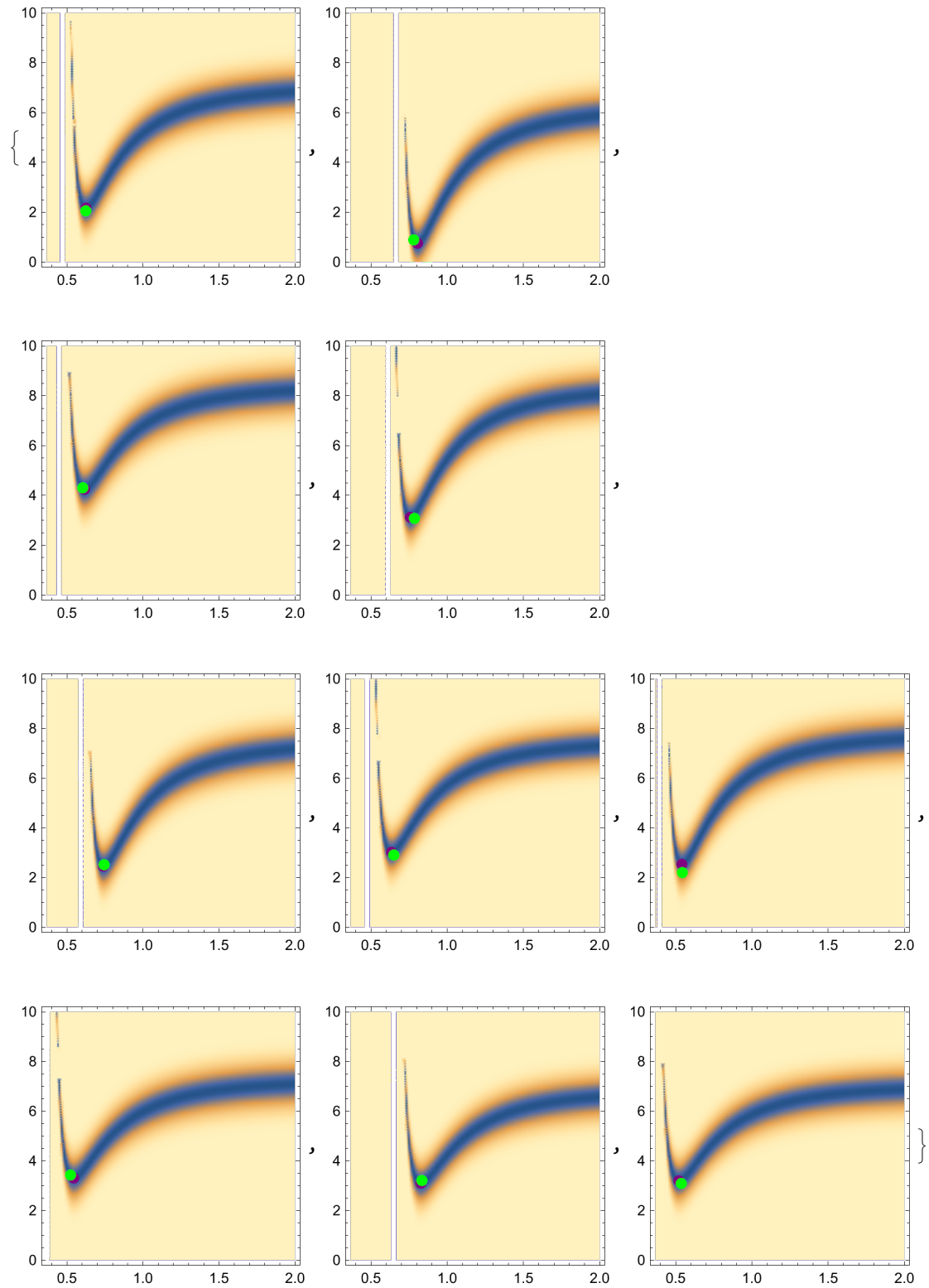
In[*]:= preas
Out[*]=
{{0.625262, 2.16867}, {0.805632, 0.753645}, {0.616186, 4.24702},
{0.758315, 3.1236}, {0.740003, 2.45962}, {0.634653, 3.01718}, {0.540726, 2.53662},
{0.544991, 3.31049}, {0.827065, 3.10368}, {0.521438, 3.19968}}
```

```

In[*]:= Table[Show[pri[[i]], ListPlot[{preas[[i]]}, PlotStyle -> {PointSize[Large], Purple}],
  ListPlot[{corpre[[i]]}, PlotStyle -> {PointSize[Large], Green}]], {i, 1, npr}]

```

Out[*]=



notebook2

```
In[*]:= nsp = 300;
npr = 10;

In[*]:= rn1 = Table[{RandomReal[], RandomReal[], RandomReal[], RandomReal[]}, {i, 1, nsp + npr}];
In[*]:= rn2 = Table[{RandomReal[], RandomReal[], RandomReal[], RandomReal[]}, {i, 1, nsp + npr}];

In[*]:= si = Table[DensityPlot[
  1 - Exp[-(1.5 + rn1[[i, 1]] 0.3) (y - (8 / 3 (0.4 + 0.2 rn1[[i, 2]] / (v - 1 / 3 - rn1[[i, 3]] / 2) -
    3 / (v - rn1[[i, 3]] / 2) ^ 2 + 6 + 3 rn1[[i, 4]])) ^ 2] + 1 -
  Exp[-(1.5 + rn2[[i, 1]] 0.3) (y - (8 / 3 (0.4 + 0.2 rn2[[i, 2]] / (v - 1 / 3 - rn2[[i, 3]] / 2) -
    3 / (v - rn2[[i, 3]] / 2) ^ 2 + 6 + 3 rn2[[i, 4]])) ^ 2],
  {v, 1 / 3 + 1 / 30, 2}, {y, 0, 10}, PlotRange -> All, PlotPoints -> 50], {i,
  1, nsp}];

In[*]:= pri = Table[DensityPlot[
  1 - Exp[-(1.5 + rn1[[i, 1]] 0.3) (y - (8 / 3 (0.4 + 0.2 rn1[[i, 2]] / (v - 1 / 3 - rn1[[i, 3]] / 2) -
    3 / (v - rn1[[i, 3]] / 2) ^ 2 + 6 + 3 rn1[[i, 4]])) ^ 2] + 1 -
  Exp[-(1.5 + rn2[[i, 1]] 0.3) (y - (8 / 3 (0.4 + 0.2 rn2[[i, 2]] / (v - 1 / 3 - rn2[[i, 3]] / 2) -
    3 / (v - rn2[[i, 3]] / 2) ^ 2 + 6 + 3 rn2[[i, 4]])) ^ 2],
  {v, 1 / 3 + 1 / 30, 2}, {y, 0, 10}, PlotRange -> All, PlotPoints -> 50], {i,
  nsp + 1, nsp + npr}];

In[*]:= cptf = Table[sol = Solve[(0.4 + 0.2 rn1[[i, 2]]) ==  $\frac{(-1 + 3 v)^2}{4 v^3}$ , v];
  {v + rn1[[i, 3]] / 2,  $\frac{-2 + 3 v}{v^3} + 6 + 3 rn1[[i, 4]]$ } /. sol[[2]], {i, 1, nsp}];

In[*]:= cpts = Table[sol = Solve[(0.4 + 0.2 rn2[[i, 2]]) ==  $\frac{(-1 + 3 v)^2}{4 v^3}$ , v];
  {v + rn2[[i, 3]] / 2,  $\frac{-2 + 3 v}{v^3} + 6 + 3 rn2[[i, 4]]$ } /. sol[[2]], {i, 1, nsp}];

In[*]:= cpt1 = Table[If[cptf[[i, 2]] < cpts[[i, 2]], cptf[[i]], cpts[[i]], {i, 1, nsp}];
In[*]:= cpt2 = Table[If[cptf[[i, 2]] > cpts[[i, 2]], cptf[[i]], cpts[[i]], {i, 1, nsp}];

In[*]:= preasf = Table[sol = Solve[(0.4 + 0.2 rn1[[i, 2]]) ==  $\frac{(-1 + 3 v)^2}{4 v^3}$ , v];
  {v + rn1[[i, 3]] / 2,  $\frac{-2 + 3 v}{v^3} + 6 + 3 rn1[[i, 4]]$ } /. sol[[2]], {i, nsp + 1, nsp + npr}];

In[*]:= preass = Table[sol = Solve[(0.4 + 0.2 rn2[[i, 2]]) ==  $\frac{(-1 + 3 v)^2}{4 v^3}$ , v];
  {v + rn2[[i, 3]] / 2,  $\frac{-2 + 3 v}{v^3} + 6 + 3 rn2[[i, 4]]$ } /. sol[[2]], {i, nsp + 1, nsp + npr}];

In[*]:= preas1 =
  Table[If[preasf[[i, 2]] < preass[[i, 2]], preasf[[i]], preass[[i]], {i, 1, Length[preasf]}]

Out[*]=
{{0.562302, 2.65756}, {0.540707, 1.82587}, {0.814673, 3.14387},
{0.717522, 4.68415}, {0.553923, 3.37838}, {0.693701, 2.72512}, {0.70964, 2.16615},
{0.618648, 4.64317}, {0.845287, 1.52144}, {0.484763, 1.25896}}
```

```
In[*]:= preas2 =
  Table[If[preasf[[i, 2]] > preass[[i, 2]], preasf[[i]], preass[[i]], {i, 1, Length[preasf]}]
Out[*]=
  {{0.640124, 4.68042}, {1.01447, 4.2048}, {0.558326, 5.60589},
   {0.933112, 5.43453}, {0.529061, 4.37608}, {0.863153, 4.16547},
   {0.631048, 4.86451}, {0.547489, 4.75672}, {0.662077, 2.28632}, {0.769785, 2.7149}}
```

```
In[*]:= datasetx1 = Table[si[[i]] → cpt1[[i, 1]], {i, 1, nsp}];
In[*]:= datasety1 = Table[si[[i]] → cpt1[[i, 2]], {i, 1, nsp}];
In[*]:= datasetx2 = Table[si[[i]] → cpt2[[i, 1]], {i, 1, nsp}];
In[*]:= datasety2 = Table[si[[i]] → cpt2[[i, 2]], {i, 1, nsp}];
In[*]:= px1 = Predict[datasetx1, Method → "GradientBoostedTrees"]
Out[*]=
```

PredictorFunction [ Input type: Image
Method: GradientBoostedTrees]

Data not saved. Save now 

```
In[*]:= py1 = Predict[datasety1, Method → "GradientBoostedTrees"]
Out[*]=
```

PredictorFunction [ Input type: Image
Method: GradientBoostedTrees]

Data not saved. Save now 

```
In[*]:= px2 = Predict[datasetx2, Method → "GradientBoostedTrees"]
Out[*]=
```

PredictorFunction [ Input type: Image
Method: GradientBoostedTrees]

Data not saved. Save now 

```
In[*]:= py2 = Predict[datasety2, Method → "GradientBoostedTrees"]
Out[*]=
```

PredictorFunction [ Input type: Image
Method: GradientBoostedTrees]

Data not saved. Save now 

```
In[*]:= corpre1 = Table[{px1[pri[[i]]], py1[pri[[i]]]}, {i, 1, Length[pri]}]
Out[*]=
  {{0.538107, 2.24495}, {0.707108, 2.47382}, {0.852711, 3.44189},
   {0.792973, 4.16849}, {0.581869, 3.73165}, {0.784609, 2.90692},
   {0.691165, 2.47294}, {0.614259, 4.25386}, {0.691955, 1.92782}, {0.64171, 1.66123}}
```

```
In[*]:= corpre2 = Table[{px2[pri[[i]]], py2[pri[[i]]]}, {i, 1, Length[pri]}]
Out[*]=
  {{0.711324, 5.02281}, {0.876775, 3.76274}, {0.581018, 5.01971},
   {0.926965, 5.50427}, {0.574904, 3.9529}, {0.883492, 4.24734},
   {0.634694, 4.15845}, {0.615109, 5.10375}, {0.835262, 2.453}, {0.796887, 2.68215}}
```

```
In[*]:= preas1
```

```
Out[*]=
```

```
{ {0.562302, 2.65756}, {0.540707, 1.82587}, {0.814673, 3.14387},  
  {0.717522, 4.68415}, {0.553923, 3.37838}, {0.693701, 2.72512}, {0.70964, 2.16615},  
  {0.618648, 4.64317}, {0.845287, 1.52144}, {0.484763, 1.25896} }
```

```
In[*]:= preas2
```

```
Out[*]=
```

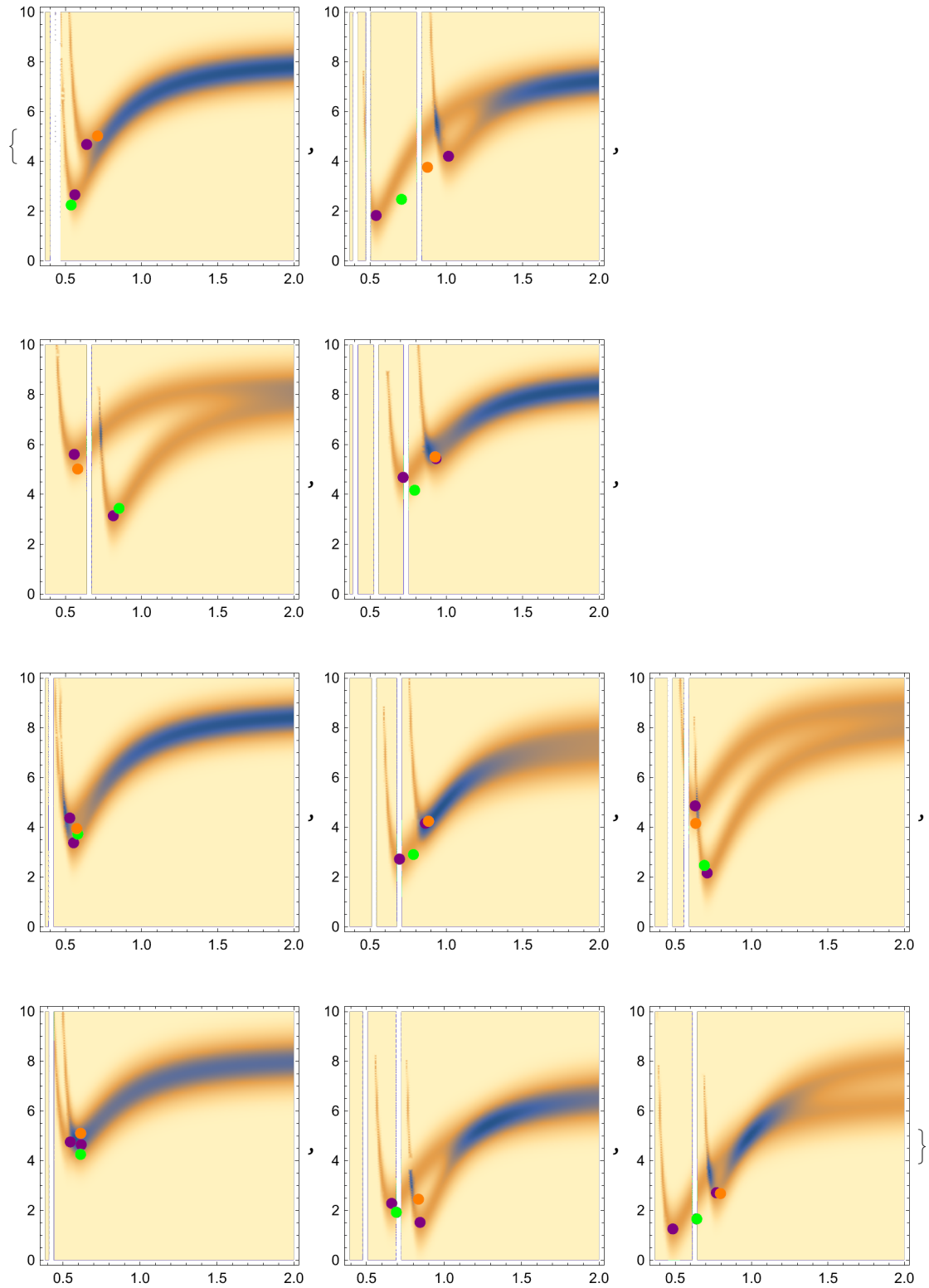
```
{ {0.640124, 4.68042}, {1.01447, 4.2048}, {0.558326, 5.60589},  
  {0.933112, 5.43453}, {0.529061, 4.37608}, {0.863153, 4.16547},  
  {0.631048, 4.86451}, {0.547489, 4.75672}, {0.662077, 2.28632}, {0.769785, 2.7149} }
```

```

In[ ]:= Table[Show[pri[[i]], ListPlot[{preas1[[i]]}, PlotStyle -> {PointSize[Large], Purple}],
  ListPlot[{preas2[[i]]}, PlotStyle -> {PointSize[Large], Purple}],
  ListPlot[{corpre1[[i]]}, PlotStyle -> {PointSize[Large], Green}],
  ListPlot[{corpre2[[i]]}, PlotStyle -> {PointSize[Large], Orange}]], {i, 1, npr}]

```

Out[]=



notebook3

```
SetDirectory[
```

```
"C:\\Users\\tsuyo\\Desktop\\MachineLearningWithWolframLanguage\\西垣\\教師データ";
```

```
In[*]:= si = {Import["T-①-1 1538.jpg"], Import["T-①-2 1541.jpg"],
  Import["T-①-3 1542.jpg"], Import["T-①-4 1543.jpg"], Import["T-①-5 1609.jpg"],
  Import["T-①-6 1622.jpg"], Import["T-①-7 1785.jpg"], Import["T-①-8 2070.jpg"],
  Import["T-①-9 2071.jpg"], Import["T-①-10 2072.jpg"], Import["T-①-11 2073.jpg"],
  Import["T-①-12 2074.jpg"], Import["T-①-13 2075.jpg"], Import["T-①-14 2076.jpg"],
  Import["T-①-15 2078.jpg"], Import["T-①-16 2079.jpg"], Import["T-①-17 2083.jpg"],
  Import["T-①-18 2319.jpg"], Import["T-①-19 2948.jpg"], Import["T-①-20 2949.jpg"],
  Import["T-①-21 3000.jpg"], Import["T-①-22 3758.jpg"], Import["T-①-23 3761.jpg"],
  Import["T-①-24 5941.jpg"], Import["T-①-25 9556.jpg"]};
```

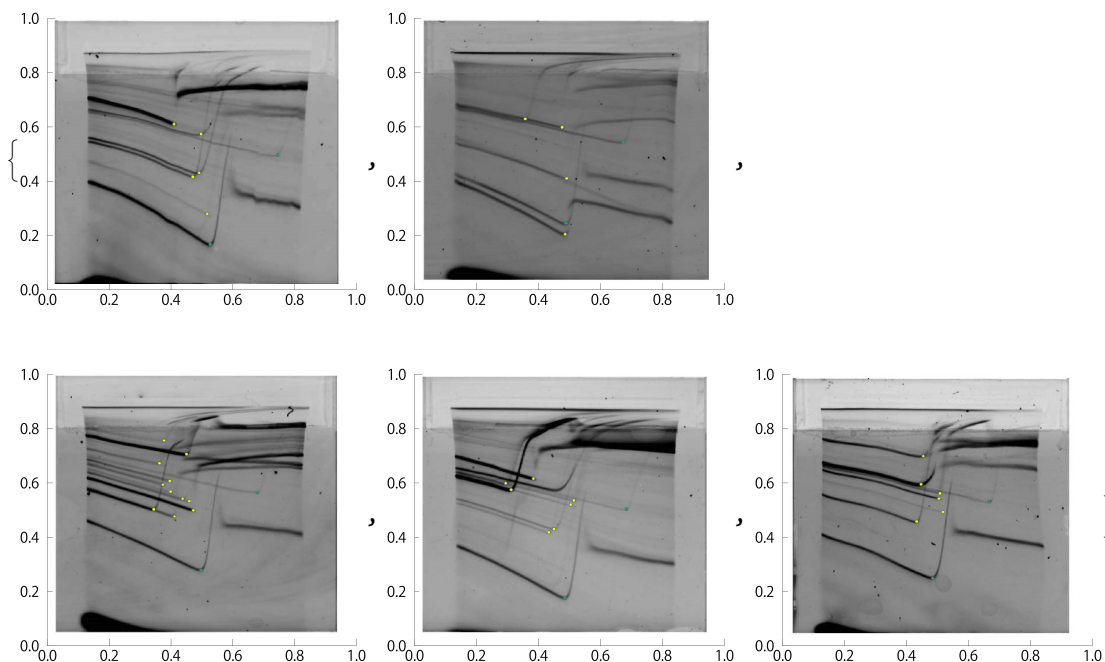
```
In[*]:= nsp = Length[si];
```

```
SetDirectory[
```

```
"C:\\Users\\tsuyo\\Desktop\\MachineLearningWithWolframLanguage\\西垣\\判定用データ";
```

```
In[*]:= pri = {Import["T-②-1 1538.jpg"], Import["T-②-2 1539.jpg"],
  Import["T-②-3 1541.jpg"], Import["T-②-4 1542.jpg"], Import["T-②-5 1543.jpg"]}
```

```
Out[*]=
```



```
In[*]:= npr = Length[pri];
```

```
In[*]:= cpt1 = {{0.544, 0.23}, {0.317, 0.2}, {0.49, 0.17}, {0.488, 0.173}, {0.488, 0.254},
  {0.51, 0.22}, {0.512, 0.271}, {0.52, 0.198}, {0.52, 0.22}, {0.485, 0.191},
  {0.472, 0.2}, {0.48, 0.215}, {0.53, 0.197}, {0.50, 0.185}, {0.528, 0.20},
  {0.483, 0.2}, {0.458, 0.209}, {0.496, 0.227}, {0.493, 0.224}, {0.52, 0.215},
  {0.53, 0.176}, {0.472, 0.229}, {0.528, 0.267}, {0.52, 0.206}, {0.507, 0.275}};
```

```
In[*]:= cpt2 = {{0.467, 0.295}, {0.216, 0.41}, {0.458, 0.266}, {0.423, 0.291}, {0.458, 0.374},
  {0.416, 0.326}, {0.685, 0.557}, {0.453, 0.289}, {0.459, 0.324}, {0.437, 0.264},
  {0.387, 0.418}, {0.419, 0.32}, {0.456, 0.315}, {0.427, 0.304}, {0.456, 0.304},
  {0.421, 0.308}, {0.427, 0.318}, {0.40, 0.427}, {0.4, 0.43}, {0.427, 0.409},
  {0.415, 0.394}, {0.376, 0.42}, {0.539, 0.459}, {0.456, 0.406}, {0.560, 0.408}};
```

```
preas1 = {{0.52, 0.165}, {0.488, 0.2}, {0.499, 0.286}, {0.483, 0.176}, {0.488, 0.251}};
preas2 = {{0.512, 0.282}, {0.491, 0.25}, {0.413, 0.477}, {0.432, 0.412}, {0.435, 0.448}};
```

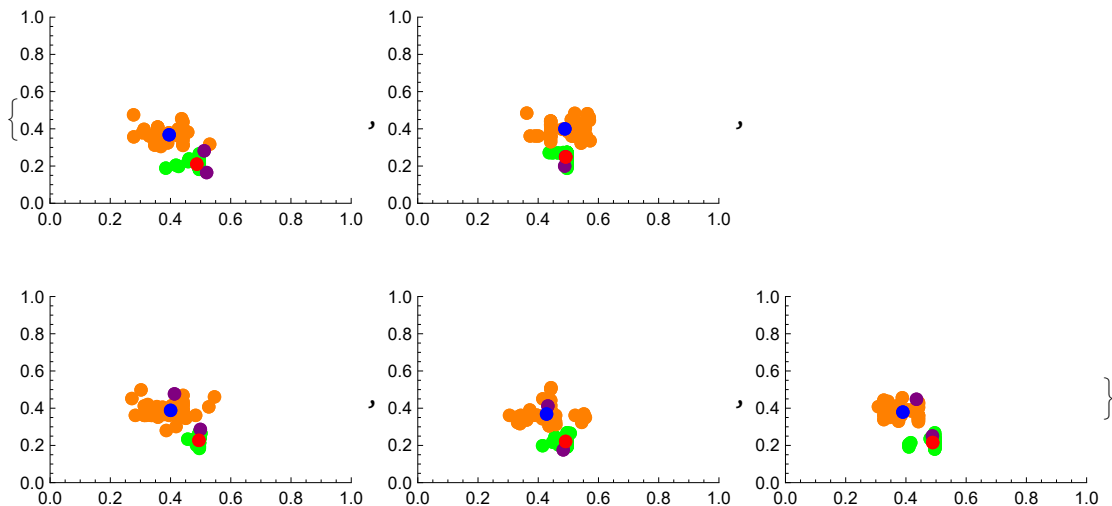
```
In[ ]:= loop = 40;
```

```
In[ ]:= tcorpre1 = {};
tcorpre2 = {};
```

```
In[ ]:= Do[datasetx1 = RandomSample[Table[si[[i]] → cpt1[[i, 1]], {i, 1, nsp}]];
datasety1 = RandomSample[Table[si[[i]] → cpt1[[i, 2]], {i, 1, nsp}]];
datasetx2 = RandomSample[Table[si[[i]] → cpt2[[i, 1]], {i, 1, nsp}]];
datasety2 = RandomSample[Table[si[[i]] → cpt2[[i, 2]], {i, 1, nsp}]];
px1 = Predict[datasetx1, Method → "GradientBoostedTrees"];
py1 = Predict[datasety1, Method → "GradientBoostedTrees"];
px2 = Predict[datasetx2, Method → "GradientBoostedTrees"];
py2 = Predict[datasety2, Method → "GradientBoostedTrees"];
corpre1 = Table[{px1[pri[[i]]], py1[pri[[i]]]}, {i, 1, Length[pri]};
corpre2 = Table[{px2[pri[[i]]], py2[pri[[i]]]}, {i, 1, Length[pri]};
tcorpre1 = AppendTo[tcorpre1, corpre1];
tcorpre2 = AppendTo[tcorpre2, corpre2], {j, 1, loop}]
```

```
In[ ]:= Table[Show[Table[ListPlot[{tcorpre1[[j, i]]},
PlotStyle → {PointSize[Large], Green}, PlotRange → {{0, 1}, {0, 1}}, {j, 1, loop}],
Table[ListPlot[{tcorpre2[[j, i]]}, PlotStyle → {PointSize[Large], Orange},
PlotRange → {{0, 1}, {0, 1}}, {j, 1, loop}], ListPlot[{preas1[[i]]},
PlotStyle → {PointSize[Large], Purple}, PlotRange → {{0, 1}, {0, 1}},
ListPlot[{preas2[[i]]}, PlotStyle → {PointSize[Large], Purple},
PlotRange → {{0, 1}, {0, 1}}, ListPlot[{Mean[tcorpre1][i]},
PlotStyle → {PointSize[Large], Red}, PlotRange → {{0, 1}, {0, 1}},
ListPlot[{Mean[tcorpre2][i]}, PlotStyle → {PointSize[Large], Blue},
PlotRange → {{0, 1}, {0, 1}}], {i, 1, npr}]
```

```
Out[ ]:=
```



```
In[ ]:= Mean[tcorpre1]
```

```
Out[ ]:=
```

```
{{0.487438, 0.21029}, {0.491612, 0.248652},
{0.494141, 0.227177}, {0.490976, 0.221461}, {0.488862, 0.215512}}
```



```
In[*]:= Mean[tcorpre2]
```

```
Out[*]=  
{ {0.395745, 0.368046}, {0.488767, 0.399858},  
  {0.400147, 0.388398}, {0.427607, 0.368624}, {0.389788, 0.379158} }
```

```
In[*]:= preas1
```

```
Out[*]=  
{ {0.52, 0.165}, {0.488, 0.2}, {0.499, 0.286}, {0.483, 0.176}, {0.488, 0.251} }
```

```
In[*]:= preas2
```

```
Out[*]=  
{ {0.512, 0.282}, {0.491, 0.25}, {0.413, 0.477}, {0.432, 0.412}, {0.435, 0.448} }
```

notebook4

SetDirectory[

"C:\\Users\\tsuyo\\Desktop\\MachineLearningWithWolframLanguage\\西垣\\教師データ";

```
In[*]:= si = {Import["T-①-1 1538.jpg"], Import["T-①-2 1541.jpg"],
  Import["T-①-3 1542.jpg"], Import["T-①-4 1543.jpg"], Import["T-①-5 1609.jpg"],
  Import["T-①-6 1622.jpg"], Import["T-①-7 1785.jpg"], Import["T-①-8 2070.jpg"],
  Import["T-①-9 2071.jpg"], Import["T-①-10 2072.jpg"], Import["T-①-11 2073.jpg"],
  Import["T-①-12 2074.jpg"], Import["T-①-13 2075.jpg"], Import["T-①-14 2076.jpg"],
  Import["T-①-15 2078.jpg"], Import["T-①-16 2079.jpg"], Import["T-①-17 2083.jpg"],
  Import["T-①-18 2319.jpg"], Import["T-①-19 2948.jpg"], Import["T-①-20 2949.jpg"],
  Import["T-①-21 3000.jpg"], Import["T-①-22 3758.jpg"], Import["T-①-23 3761.jpg"],
  Import["T-①-24 5941.jpg"], Import["T-①-25 9556.jpg"]};
```

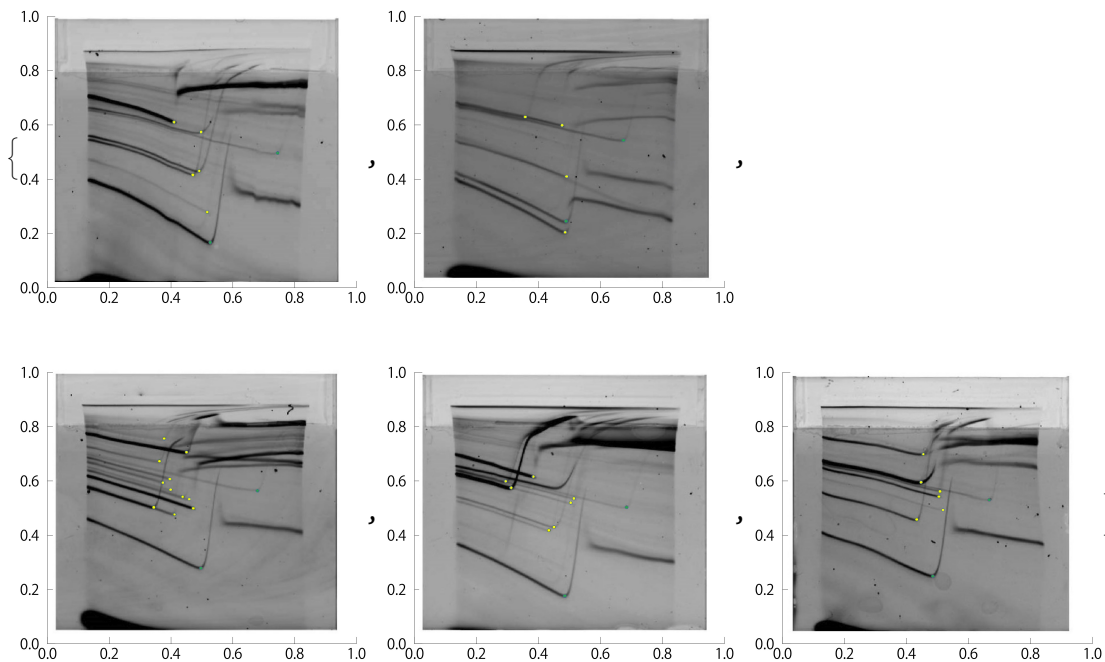
```
In[*]:= nsp = Length[si];
```

SetDirectory[

"C:\\Users\\tsuyo\\Desktop\\MachineLearningWithWolframLanguage\\西垣\\判定用データ";

```
In[*]:= pri = {Import["T-②-1 1538.jpg"], Import["T-②-2 1539.jpg"],
  Import["T-②-3 1541.jpg"], Import["T-②-4 1542.jpg"], Import["T-②-5 1543.jpg"]}
```

Out[*]=



```
In[*]:= npr = Length[pri];
```

```
In[*]:= cpt1 = {{0.52, 0.455}, {0.51, 0.51}, {0.445, 0.39}, {0.397, 0.45}, {0.53, 0.66},
  {0.48, 0.468}, {0.413, 0.628}, {0.71, 0.477}, {0.728, 0.518}, {0.387, 0.435},
  {0.664, 0.523}, {0.37, 0.488}, {0.405, 0.470}, {0.373, 0.457}, {0.507, 0.44},
  {0.467, 0.44}, {0.469, 0.456}, {0.680, 0.536}, {0.677, 0.534}, {0.707, 0.515},
  {0.731, 0.489}, {0.672, 0.503}, {0.709, 0.556}, {0.723, 0.487}, {0.525, 0.48}};
```

```
In[*]:= cpt2 = {{0.405, 0.47}, {0.227, 0.56}, {0.397, 0.42}, {0.41, 0.49}, {0.53, 0.66},
  {0.68, 0.47}, {0.51, 0.668}, {0.432, 0.491}, {0.445, 0.520}, {0.472, 0.447},
  {0.416, 0.568}, {0.461, 0.52}, {0.723, 0.509}, {0.46, 0.490}, {0.395, 0.445},
  {0.36, 0.457}, {0.363, 0.465}, {0.44, 0.57}, {0.44, 0.567}, {0.432, 0.565},
  {0.42, 0.516}, {0.389, 0.553}, {0.709, 0.556}, {0.723, 0.487}, {0.677, 0.541}};
```

```

In[ ]:= preas1 = {{0.747, 0.476}, {0.480, 0.588}, {0.461, 0.526}, {0.501, 0.481}, {0.48, 0.518}}
Out[ ]:= {{0.747, 0.476}, {0.48, 0.588}, {0.461, 0.526}, {0.501, 0.481}, {0.48, 0.518}}

In[ ]:= preas2 = {{0.493, 0.532}, {0.355, 0.618}, {0.44, 0.537}, {0.515, 0.528}, {0.512, 0.535}}
Out[ ]:= {{0.493, 0.532}, {0.355, 0.618}, {0.44, 0.537}, {0.515, 0.528}, {0.512, 0.535}}

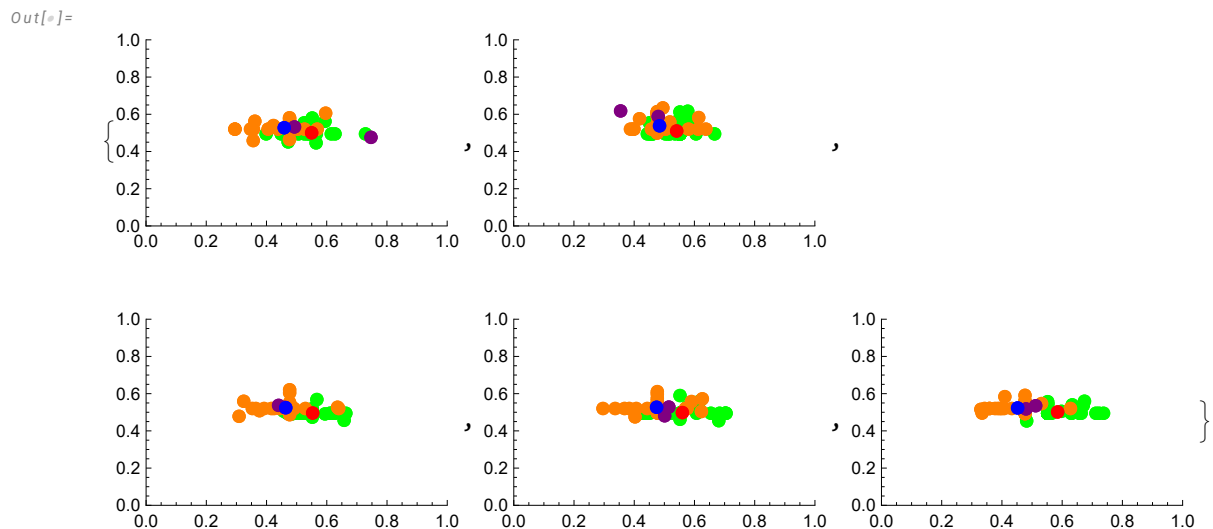
In[ ]:= loop = 40;

In[ ]:= tcorpre1 = {};
tcorpre2 = {};

In[ ]:= Do[datasetx1 = RandomSample[Table[si[[i]] → cpt1[[i, 1]], {i, 1, nsp}]];
datasety1 = RandomSample[Table[si[[i]] → cpt1[[i, 2]], {i, 1, nsp}]];
datasetx2 = RandomSample[Table[si[[i]] → cpt2[[i, 1]], {i, 1, nsp}]];
datasety2 = RandomSample[Table[si[[i]] → cpt2[[i, 2]], {i, 1, nsp}]];
px1 = Predict[datasetx1, Method → "GradientBoostedTrees"];
py1 = Predict[datasety1, Method → "GradientBoostedTrees"];
px2 = Predict[datasetx2, Method → "GradientBoostedTrees"];
py2 = Predict[datasety2, Method → "GradientBoostedTrees"];
corpre1 = Table[{px1[pri[[i]]], py1[pri[[i]]]}, {i, 1, Length[pri]}];
corpre2 = Table[{px2[pri[[i]]], py2[pri[[i]]]}, {i, 1, Length[pri]}];
tcorpre1 = AppendTo[tcorpre1, corpre1];
tcorpre2 = AppendTo[tcorpre2, corpre2], {j, 1, loop}]

In[ ]:= Table[Show[Table[ListPlot[{tcorpre1[[j, i]]},
PlotStyle → {PointSize[Large], Green}, PlotRange → {{0, 1}, {0, 1}}, {j, 1, loop}],
Table[ListPlot[{tcorpre2[[j, i]]}, PlotStyle → {PointSize[Large], Orange},
PlotRange → {{0, 1}, {0, 1}}, {j, 1, loop}], ListPlot[{preas1[[i]]},
PlotStyle → {PointSize[Large], Purple}, PlotRange → {{0, 1}, {0, 1}},
ListPlot[{preas2[[i]]}, PlotStyle → {PointSize[Large], Purple},
PlotRange → {{0, 1}, {0, 1}}, ListPlot[{Mean[tcorpre1][[i]]},
PlotStyle → {PointSize[Large], Red}, PlotRange → {{0, 1}, {0, 1}},
ListPlot[{Mean[tcorpre2][[i]]}, PlotStyle → {PointSize[Large], Blue},
PlotRange → {{0, 1}, {0, 1}}]], {i, 1, npr}]

```



```
In[ ]:= Mean[tcorpre1]
```

```
Out[ ]:=
{{0.549958, 0.500318}, {0.541167, 0.510241},
 {0.553063, 0.495854}, {0.559854, 0.498205}, {0.584848, 0.50166}}
```

```
In[ ]:= Mean[tcorpre2]
```

```
Out[ ]:=
{{0.458923, 0.5278}, {0.484308, 0.537956},
 {0.463793, 0.52436}, {0.474176, 0.526961}, {0.45204, 0.523241}}
```

```
In[ ]:= preas1
```

```
Out[ ]:=
{{0.747, 0.476}, {0.48, 0.588}, {0.461, 0.526}, {0.501, 0.481}, {0.48, 0.518}}
```

```
In[ ]:= preas2
```

```
Out[ ]:=
{{0.493, 0.532}, {0.355, 0.618}, {0.44, 0.537}, {0.515, 0.528}, {0.512, 0.535}}
```

notebook5

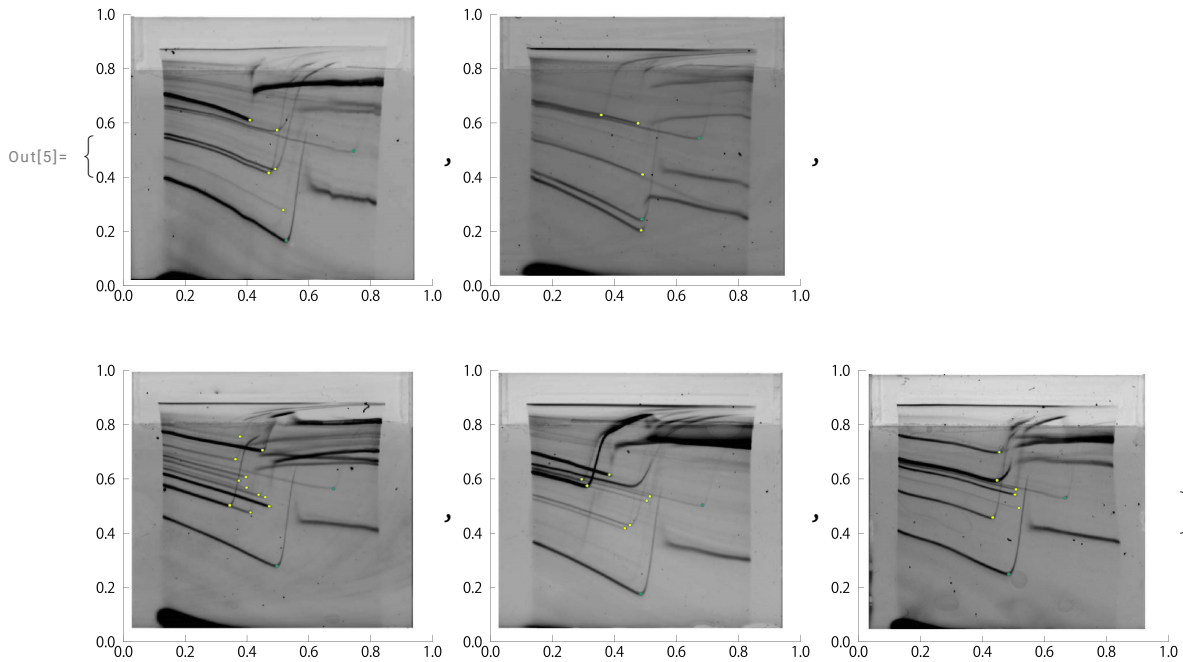
```
In[1]:= SetDirectory[  
    "C:\\Users\\tsuyo\\Desktop\\MachineLearningWithWolframLanguage\\西垣\\教師データ";
```

```
In[2]:= si = {Import["T-㊦-1 1538.jpg"], Import["T-㊦-2 1541.jpg"],  
    Import["T-㊦-3 1542.jpg"], Import["T-㊦-4 1543.jpg"], Import["T-㊦-5 1609.jpg"],  
    Import["T-㊦-6 1622.jpg"], Import["T-㊦-7 1785.jpg"], Import["T-㊦-8 2070.jpg"],  
    Import["T-㊦-9 2071.jpg"], Import["T-㊦-10 2072.jpg"], Import["T-㊦-11 2073.jpg"],  
    Import["T-㊦-12 2074.jpg"], Import["T-㊦-13 2075.jpg"], Import["T-㊦-14 2076.jpg"],  
    Import["T-㊦-15 2078.jpg"], Import["T-㊦-16 2079.jpg"], Import["T-㊦-17 2083.jpg"],  
    Import["T-㊦-18 2319.jpg"], Import["T-㊦-19 2948.jpg"], Import["T-㊦-20 2949.jpg"],  
    Import["T-㊦-21 3000.jpg"], Import["T-㊦-22 3758.jpg"], Import["T-㊦-23 3761.jpg"],  
    Import["T-㊦-24 5941.jpg"], Import["T-㊦-25 9556.jpg"]};
```

```
In[3]:= nsp = Length[si];
```

```
In[4]:= SetDirectory[  
    "C:\\Users\\tsuyo\\Desktop\\MachineLearningWithWolframLanguage\\西垣\\判定用データ";
```

```
In[5]:= pri = {Import["T-㊧-1 1538.jpg"], Import["T-㊧-2 1539.jpg"],  
    Import["T-㊧-3 1541.jpg"], Import["T-㊧-4 1542.jpg"], Import["T-㊧-5 1543.jpg"]}
```



```
In[6]:= npr = Length[pri];
```

```

In[7]:= cpt = {{ {0.544`, 0.23`}, {0.467`, 0.295`},
  {0.49`, 0.415`}, {0.43`, 0.45`}, {0.52`, 0.455`}, {0.405`, 0.47`}},
  {{0.317`, 0.2`}, {0.216`, 0.41`}, {0.216`, 0.45`}, {0.285`, 0.457`},
  {0.51`, 0.51`}, {0.227`, 0.56`}}, {{0.49`, 0.17`}, {0.458`, 0.266`},
  {0.437`, 0.28`}, {0.432`, 0.33`}, {0.445`, 0.39`}, {0.397`, 0.42`}},
  {{0.488`, 0.173`}, {0.423`, 0.291`}, {0.397`, 0.45`}, {0.648`, 0.486`},
  {0.397`, 0.45`}, {0.41`, 0.49`}}, {{0.488`, 0.254`}, {0.458`, 0.374`},
  {0.51`, 0.477`}, {0.656`, 0.54`}, {0.53`, 0.66`}, {0.53`, 0.66`}},
  {{0.51`, 0.22`}, {0.416`, 0.326`}, {0.41`, 0.41`}, {0.408`, 0.46`},
  {0.48`, 0.468`}, {0.68`, 0.47`}}, {{0.512`, 0.271`}, {0.685`, 0.557`},
  {0.461`, 0.594`}, {0.477`, 0.626`}, {0.413`, 0.628`}, {0.51`, 0.668`}},
  {{0.52`, 0.198`}, {0.453`, 0.289`}, {0.496`, 0.431`}, {0.395`, 0.44`},
  {0.71`, 0.477`}, {0.432`, 0.491`}}, {{0.52`, 0.22`}, {0.459`, 0.324`},
  {0.467`, 0.426`}, {0.408`, 0.465`}, {0.728`, 0.518`}, {0.445`, 0.52`}},
  {{0.485`, 0.191`}, {0.437`, 0.264`}, {0.402`, 0.303`}, {0.437`, 0.412`},
  {0.387`, 0.435`}, {0.472`, 0.447`}}, {{0.472`, 0.2`}, {0.387`, 0.418`},
  {0.373`, 0.471`}, {0.445`, 0.482`}, {0.664`, 0.523`}, {0.416`, 0.568`}},
  {{0.48`, 0.215`}, {0.419`, 0.32`}, {0.44`, 0.435`}, {0.469`, 0.473`},
  {0.37`, 0.488`}, {0.461`, 0.52`}}, {{0.53`, 0.197`}, {0.456`, 0.315`},
  {0.475`, 0.42`}, {0.507`, 0.461`}, {0.405`, 0.47`}, {0.723`, 0.509`}},
  {{0.5`, 0.185`}, {0.427`, 0.304`}, {0.44`, 0.409`}, {0.469`, 0.45`},
  {0.373`, 0.457`}, {0.46`, 0.49`}}, {{0.528`, 0.2`}, {0.456`, 0.304`},
  {0.469`, 0.4`}, {0.413`, 0.435`}, {0.507`, 0.44`}, {0.395`, 0.445`}},
  {{0.483`, 0.2`}, {0.421`, 0.308`}, {0.432`, 0.409`}, {0.389`, 0.437`},
  {0.467`, 0.44`}, {0.36`, 0.457`}}, {{0.458`, 0.209`}, {0.427`, 0.318`},
  {0.437`, 0.415`}, {0.387`, 0.45`}, {0.469`, 0.456`}, {0.363`, 0.465`}},
  {{0.496`, 0.227`}, {0.4`, 0.427`}, {0.395`, 0.475`}, {0.467`, 0.488`},
  {0.68`, 0.536`}, {0.44`, 0.57`}}, {{0.493`, 0.224`}, {0.4`, 0.43`},
  {0.392`, 0.475`}, {0.467`, 0.487`}, {0.677`, 0.534`}, {0.44`, 0.567`}},
  {{0.52`, 0.215`}, {0.427`, 0.409`}, {0.416`, 0.459`}, {0.493`, 0.47`},
  {0.707`, 0.515`}, {0.432`, 0.565`}}, {{0.53`, 0.176`}, {0.415`, 0.394`},
  {0.433`, 0.439`}, {0.492`, 0.45`}, {0.731`, 0.489`}, {0.42`, 0.516`}},
  {{0.472`, 0.229`}, {0.376`, 0.42`}, {0.379`, 0.462`}, {0.453`, 0.47`},
  {0.672`, 0.503`}, {0.389`, 0.553`}}, {{0.528`, 0.267`}, {0.539`, 0.459`},
  {0.472`, 0.488`}, {0.709`, 0.556`}, {0.709`, 0.556`}, {0.709`, 0.556`}},
  {{0.52`, 0.206`}, {0.456`, 0.406`}, {0.496`, 0.464`}, {0.475`, 0.487`},
  {0.723`, 0.487`}, {0.723`, 0.487`}}, {{0.507`, 0.275`}, {0.56`, 0.408`},
  {0.559`, 0.432`}, {0.528`, 0.459`}, {0.525`, 0.48`}, {0.677`, 0.541`}}};

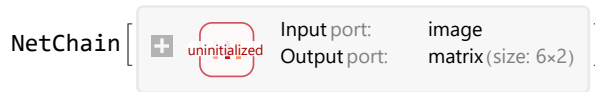
In[8]:= preas = {{ {0.52`, 0.165`}, {0.512`, 0.282`},
  {0.467`, 0.406`}, {0.488`, 0.421`}, {0.747`, 0.476`}, {0.405`, 0.47`}},
  {{0.488`, 0.2`}, {0.491`, 0.25`}, {0.491`, 0.406`}, {0.68`, 0.541`},
  {0.48`, 0.588`}, {0.227`, 0.56`}}, {{0.499`, 0.286`}, {0.413`, 0.477`},
  {0.475`, 0.494`}, {0.347`, 0.497`}, {0.461`, 0.526`}, {0.397`, 0.42`}},
  {{0.483`, 0.176`}, {0.432`, 0.412`}, {0.451`, 0.427`}, {0.685`, 0.468`},
  {0.501`, 0.481`}, {0.41`, 0.49`}}, {{0.488`, 0.251`}, {0.435`, 0.448`},
  {0.52`, 0.479`}, {0.672`, 0.51`}, {0.48`, 0.518`}, {0.53`, 0.66`}}};

In[9]:= dataset = Table[si[[i]] -> cpt[[i]], {i, 1, nsp}];

```

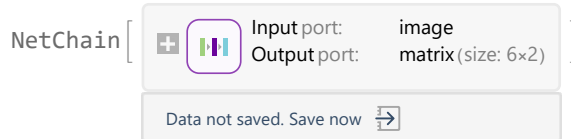
```
In[10]:= net = NetChain[{LinearLayer[10], LinearLayer[5],
  LinearLayer[7], LinearLayer[{6, 2}]], "Input" → "Image"]
```

Out[10]=



```
In[11]:= trained = NetTrain[net, dataset]
```

Out[11]=



```
In[12]:= sol = {trained[pri[[1]], trained[pri[[2]],
  trained[pri[[3]], trained[pri[[4]], trained[pri[[5]]]}
```

Out[12]=

```
{{{0.487467, 0.186928}, {0.444653, 0.359699}, {0.437278, 0.430796},
  {0.498015, 0.447976}, {0.569141, 0.515786}, {0.480482, 0.491373}},
 {{0.589463, 0.213595}, {0.468189, 0.30134}, {0.458126, 0.483195},
  {0.497773, 0.429016}, {0.580543, 0.562811}, {0.475682, 0.575215}},
 {{0.573034, 0.265292}, {0.467506, 0.291415}, {0.508128, 0.468082},
  {0.426595, 0.49193}, {0.581408, 0.43913}, {0.461256, 0.524992}},
 {{0.580782, 0.261814}, {0.400881, 0.272158}, {0.499576, 0.521257},
  {0.485136, 0.564059}, {0.547241, 0.417673}, {0.418919, 0.500074}},
 {{0.491168, 0.212063}, {0.439503, 0.419823}, {0.424286, 0.449598},
  {0.501483, 0.46661}, {0.592642, 0.499286}, {0.51165, 0.545129}}}
```

```
In[13]:= Table[ListPlot[{sol[[i]], preas[[i]]}, AspectRatio → 1,
  PlotRange → {{0, 1}, {0, 1}}, PlotStyle → PointSize[Large]], {i, 1, 5}]
```

Out[13]=

