

# 学習者用デジタル教科書ビューアにおける ドラッグ操作自動判別手法の開発

小島江里加<sup>1</sup> 加藤直樹<sup>2</sup>

**概要:** 学習者用デジタル教科書ビューアにおける書き込みとページめくりはともにドラッグ操作であることが多く、その区別のためのモード切替を失念することによる誤操作がユーザビリティを悪化させている。この問題は、誤操作を自動で修正する機能によって解決できると考える。本稿では、誤操作修正機能実装の前段階として、ドラッグ操作時に端末から取得できる運筆データの特徴量とする判別モデルによって、ドラッグ操作が書き込みとページめくりのどちらを意図したものであるかを判別する手法の開発について述べる。判別モデルの性能評価の結果、ドラッグ操作初動時に得られる運筆データを用いて判別するモデルではおおむね判別可能であり、ドラッグ操作終了時まで得られる運筆データを用いて判別するモデルでは高精度に判別可能であることがわかった。

**キーワード:** 学習者用デジタル教科書ビューア, ドラッグ操作, 誤操作修正, 自動判別, 機械学習

## Developing an Automated Method for Discriminating Drag Operations in Digital Textbooks for Learners

ERIKA KOJIMA<sup>†1</sup> NAOKI KATO<sup>†2</sup>

### 1. はじめに

GIGA スクール構想によって、教育現場の ICT (Information and Communication Technology) 環境整備が加速した。1 人 1 台端末、学習者用デジタル教科書を利用できる環境が整い、今後ますます学習者用デジタル教科書・教材を効果的に活用し、より高い学習効果を生み出していくことが求められている。

ここで、学習者用デジタル教科書とは、紙の教科書をデジタル化したデータと定義されている。一般にはこのデータを表示し、スタイラスペン（以下、ペン）による書き込みや拡大・縮小などの機能を提供するソフトウェア（デジタル教科書ビューア、以下ビューアと記す）と合わせて「学習者用デジタル教科書」と呼んでいる。さらに、教科書に関連するデジタル教材を含めたものを「学習者用デジタル教科書・教材」等と呼ぶ。本論文では、これら「学習者用デジタル教科書」「学習者用デジタル教科書・教材」を区別せずに「学習者用デジタル教科書・教材」と記す。

ビューアにおいて、書き込みは紙面上でドラッグ操作（ペン入力の場合はペン先を画面に下ろしてそのまますべらせ、タッチ入力の場合は指で画面をタッチしたまま指をすべらせる）することで行える。ページ移動の方法はビューアによって異なるが、1 ページごとのページ移動（以下、ページめくり）をドラッグ操作で行えるビューアが多い。

そのため、ドラッグ操作が書き込みなのかページめく

りなのかを切り替えるためにモードを取り入れている。ページめくりモードのときにドラッグするとページめくりが、書き込みモードのときにドラッグすると書き込みが行える。その二つのモードの切替は、通常、ユーザがモード切替用ボタンを押すことで行う。

このようなモーダルなインタフェースでは、モードの切り替えを失念することで、意図した操作が行えないことがある。たとえば、ページめくりを行おうとしたときに、書き込みモードであることを失念してモード切替用ボタンを押さずにドラッグすると、ページめくりではなく書き込みが行われてしまう。同様に、書き込みを行おうとしたがページめくりが行われてしまうこともある。こうした誤操作によって、何を書き込もうとしたのか忘れてしまったり、操作に注意が向き教員の話の聞き逃してしまったりと、直接的に学習に支障をきたす。また、誤操作発生時の不快感情によって後の学習に集中できなくなってしまうなど、間接的に学習に支障をきたすことも考えられる。

これらのモーダルなインタフェースが抱える問題は、ユーザがモード切替を失念していることにシステムが気づき、ページをめくろうとしたが書き込みが行われてしまった場合には、既に書き込まれたものを消してページがめくられた状態にし、書き込もうとしたがページめくりが行われてしまった場合には、ページを戻して、本来書かれるはずだった書き込みも書かれた状態にするといった誤操作修正機能によって解決できると考える。

そこで本研究では、学習者用デジタル教科書・教材のユーザビリティを向上し、学習者用デジタル教科書・教材の活用による学習効果をより高められるようにすることを目標とし、上記の機能実装の前段階として、学習者用デジ

<sup>1</sup> 東京学芸大学教育学部  
Faculty of Education, Tokyo Gakugei University  
<sup>2</sup> 東京学芸大学  
Tokyo Gakugei University

タル教科書ビューアにおけるドラッグ操作がページめくりと書き込みのどちらを行おうとしたものなのかを、端末から取得できるデバイス入力情報を使用して判別する手法を検討し、その実装と評価を通して、有用性を検証することを目的とする。

## 2. 関連研究

### 2.1 誤り修正に関する研究

本研究では、誤操作修正機能の実現を目指す、誤り修正のアプローチはさまざまである[1][2][3][4]。

誤り修正には誤り箇所のみを指摘するもの、誤りの内容を提示するもの、修正案を提示するもの、誤りを自動で修正するものなどがある。どのような誤りを犯しているかを考えさせることが重要な場合は誤り箇所の提示のみを行い、単純な入力ミスであれば自動で修正するなど、場面によって適切な誤り修正のアプローチは異なる。本研究で扱う、モードの切り替え忘れによる誤操作は、単に操作手順の一つを行い忘れたことによるものであり、誤りがあったことをユーザが知る必要はないため、自動で修正すべきである。

### 2.2 タッチ入力語彙拡張に関する研究

タッチインタフェースにおいて、より多様な入力方法の実現によってユーザビリティを向上しようとする入力語彙拡張手法の提案やそれに特定の操作を割り当てようとする研究が行われている[5][6][7][8]。

書き込みモードとページめくりモードにそれぞれ異なる操作を割り当てることも考えられるが、これまでと異なる操作方法になることで新たな誤操作が生じる可能性がある。そのため、本研究では学習環境を変えずに誤操作を防ぐことを目指す。

### 2.3 行動推定(状態判別)に関する研究

本研究では、学習者用デジタル教科書・教材において、ドラッグ操作が「書き込み」と「ページめくり」のどちらを意図したものであるかを推定する。行動推定や状態判別を試みる研究は数多く行われており、推定にはさまざまな情報が用いられている。

藤好らは、英文問題解答時の視点情報から英語習熟度を推定する手法を提案した[9]。ドラッグ操作が行われている位置と視点が一致している場合は書き込みを行っている可能性が高いといったことが予想され、視点情報は本研究においても有効であると思われる。しかし、視点情報の取得にはアイトラッカーなどの機器が必要となるため現在の教育現場での使用は現実的でない。

各務らは、ユーザビリティ評価の支援を目的とし、ユーザビリティテスト時の音声情報から、機器操作時に感じるとまどいを推定するシステムを開発した[10]。学習者用デジタル教科書・教材使用時にも、誤操作時には「あっ」「間違えた」といった発言が予想され、音声情報は有用で

あるように思われる。しかし、主に学習者用デジタル教科書・教材が使用される授業場面は、教員が話し続けていたり、周りにいる児童・生徒の音声が入ったりと、ユーザ本人以外の音声が入りやすい環境である。また、音声を常に取得できるようにしておくことはデバイスに負担がかかるため、処理速度等の点で学習に支障をきたす可能性がある。

江木らは、筆記具に取り付けた加速度センサを用いて学習者の状態を「筆記具を置いている」「筆記具を持った手が止まっている」「筆記具を持って動かしている」「筆記具を使って書いている」に分類する手法を検討し、加速度センサで得られる情報から状態推定が可能であることを示した[11]。高橋らは、タブレット端末で取得可能なオンライン筆記データを使用して、作文における「書き写し行為」と「創作的行為」を推定する手法を提案した[12]。このような、標準的な入力デバイスから取得可能な筆記データを用いる場合、追加のデバイスが不要なため、教育現場においても利用しやすいデータとなる。本研究でもデバイスから取得できる筆記データを用いたドラッグ操作自動判別手法を検討する。

## 3. 判別に用いるデータの収集

### 3.1 運筆データ

#### 3.1.1 運筆データ候補の選定

ここでは、書き込みとページめくりの判別方式を探求する上で、判別に有用と考えられるデータ(以下、運筆データと呼ぶ)の選定について述べる。

ドラッグ操作は一本の線(ストローク)を描く動作に相当し、その形状は不連続な点の集合として表現される。この点の座標(列)は運筆データの基本となる。書き込みは余白のある紙面の縁付近で行い、ページめくりは紙面の中央付近で行うと予想されるため、座標はこの位置の区別にも使える。

また、ページめくりのドラッグ操作は横線を引くような動作であり、ドラッグ操作の縦幅は小さくなるが、書き込みでは縦線を書くこともあり、その場合、縦幅は大きくなる。縦幅が大きければ書き込みであると判断できると考えられる。加えて、余白などの限られたスペースに書くことの多い書き込みは比較的小さくドラッグ操作を行い、その点を気にしなくてよいページめくりでは比較的大きくドラッグ操作を行うと予想される。このドラッグ操作の大きさの違いはドラッグ操作の横幅にもあらわれる。そこで、ストローク外接矩形(ストロークを表現するすべての点を内包し、各辺がX軸またはY軸に並行な最小の長方形)の縦幅と横幅(これらをストローク外接矩形幅と呼ぶ)を運筆データとする。

ストローク全体の筆記速度、筆圧、加速度、ストローク間の時間間隔から筆記者の行為推定が可能[12]であることから、本研究においてもこれらのデータが有用である可

能性がある。Yuらは、認知的負荷の大きさと筆記動作との関係を調査し、筆圧と筆記速度が認知的負荷を評価する指標として有効であることを示した[13]。学習者用デジタル教科書・教材において、ページめくりは横線を引くような単純な動作であるのに対し、書き込みは書き込む内容を整理したり、書き込む位置を考えたりする必要がある。そのため、この二つの行動では、認知的負荷の大きさに差が出ると考えられることから、本研究において筆圧や筆記速度が判別に有効であると考えられる。

筆記速度は変位を単位時間で割るため、ドラッグ操作開始直後などでは筆記速度は非常に小さい値になると考えられる。そこで、比較的大きく、似た要素である変位も運筆データとする。

文字を書くときの要素として「とめ・はね・はらい」がある。書き込みはこの3要素すべてから構成されるが、ページめくりは「はらい」に近い動作である。「はらい」の動作では、加速度には正の値が多く出現し「とめ」の動作では止めようとする力がはたらくため、加速度には負の値が出現しやすいと予想されることから、加速度は有効な運筆データとして適当であると考えられる。

西村らは、タッチパネルにおいて始点から目標点までの、指先を使用したドラッグ操作におけるピーク速度出現時間を調べた[14]。1回のドラッグ操作にかかった時間を100%としたとき、ピーク速度はドラッグ操作開始後約10~20%時点で出現することがわかっている。この知見からは、書き込みとページめくりにおいても加速度の違いが特にあらわれるのはドラッグ操作開始後20%時点以降（ストロークの残り4/5以降）であると考えられる。しかし、簡易的な調査を行ったところ、止めようとする力がはたらくのはストロークの後半部分であるとイメージする人が多かった。そこで、ストロークの残り4/5以降、2/5以降、1/5以降の加速度を運筆データとする。

少し離れたページを開くときには、同じ方向に連続してページめくりを行う。この場合、1回目のページめくりの後、手を元の位置に戻してから2回目のページめくりを行うと考える。そうだとすれば、書き込みはほぼ同じ位置に連続してドラッグ操作を行うが、ページめくりは手の位置を動かすため、ページめくりよりも書き込みの方が、前回のドラッグ操作が終了してから次のドラッグ操作を開始するまでの経過時間は短くなると考える。この経過時間を運筆データとする。

塗りつぶしを行う場合、一筆で塗りつぶすことも多く、その場合、ドラッグ操作にかかる時間は長くなる。ドラッグ時間が長い場合は書き込みの可能性が高いと考えられるため、ドラッグ時間を運筆データとする。

以上の関連研究より得られる知見および経験に基づく推測から、本研究では座標、変位、速度、加速度、筆圧、経過時間、ドラッグ時間、ストローク外接矩形幅を運筆デ

ータとし、判別の要素として検討する。

### 3.1.2 判別タイミングと、それごとに使用する運筆データ

ここでは、前節で選定した候補からの判別タイミングごとに使用する運筆データについて述べる。

学習者用デジタル教科書・教材において、ドラッグ操作の特徴からユーザが意図する操作を判別するタイミングとして次の二つが考えられる。

一つ目は、ドラッグ操作をし始めた瞬間に判別を行う方法である。この方法では、ドラッグ操作開始直後まで（初動）の運筆データを使用して判別し、自動でモード切替を行うことで、モード切替を失念した場合でも意図した操作を行えるようになる。二つ目は、ドラッグ操作終了時に判別を行う方法である。この方法では、ドラッグ操作終了時までに得られる運筆データから判別を行うことで、誤操作を修正できる。

ドラッグ操作をし始めた瞬間に判別するための要素として、ドラッグ操作初動時に得られる座標、変位、速度、加速度、筆圧、経過時間を用いる。これらの運筆データを、以下、初動データと呼ぶ。

ドラッグ操作終了時に判別するための要素として、ドラッグ操作終了時までに得られるドラッグ時間、ストローク外接矩形幅と、速度、加速度、筆圧については1ストローク内の平均、中央値、最大値、最小値、最初と最後に取れるデータを取得する。加速度はストロークの残り4/5以降、2/5以降、1/5以降の加速度も取得する。これらの運筆データを、以下、終了時データと呼ぶ。

変位、速度、加速度は、X成分、Y成分も取得する。

### 3.1.3 物理座標系と紙面座標系

変位、速度、加速度、ストローク外接矩形幅については、物理座標系における値と紙面座標系における値の2種類取得する（図1）。物理座標系と紙面座標系について、変位を例に説明する。物理座標系はタッチパネルの画面の左上が原点となる座標系で、この座標系における変位は実際の指やペンの動きに合った変位である。一方の紙面座標系はデジタル教科書紙面の左上が原点となる座標系で、この座標系における変位は紙面上における変位をあらわし、表示の拡大・縮小に依存する。たとえば、拡大・縮小を行っていないときと2倍に拡大したときでそれぞれ同じ長さの線を描くとする。このとき、物理座標系の変位は同じ値となる。一方、紙面座標系の変位は、拡大しているときは拡大していないときの1/2の値になる。

なお、物理座標系における変位、速度、加速度、ストローク外接矩形幅については、今回用いた実行環境以外のデバイスを使う場合、1pxの大きさが変わることがあるため、その補正、たとえば、縦横方向それぞれについて画素数が同じでサイズが2倍の場合は1pxの大きさが2倍になるので、入力データを物理座標系に変換する際には2倍にする。

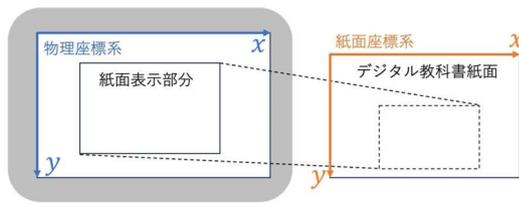


図1 物理座標系と紙面座標系

## 3.2 データ収集環境

### 3.2.1 ハードウェア環境

データの収集には、IPS 液晶（静電タッチパネル付き）を搭載した Windows PC（Panasonic 製、Let's note CF-MX5）を用い、タブレットモードで使用してもらった。画面のサイズは 12.5inch（約 277mm×156mm）、表示画素数は 1920px×1080px である。本研究ではこのデバイスだけを利用することを前提とし、物理座標系の単位には、簡易的に、絶対単位である mm ではなく、画面座標の単位の px を用いる。

ペンを使って行うタスクでは、デバイス付属のスタイラスペンを使用した。

### 3.2.2 ソフトウェア環境

データ収集用に、ドラッグ操作によるページめくり機能、フリーハンドで書き込める書き込み機能、取得した運筆データを CSV 形式のデータファイルでダウンロードできるデータ取得機能を持つ簡易的なビューア（以下、データ取得用デジタル教科書）を開発した。

開発言語は JavaScript, HTML, CSS を使用し、Google Chrome を使用して動作確認を行った。筆圧の取得には、JavaScript の TouchEvent オブジェクトの force プロパティを使用した。拡大・縮小はブラウザに備わっている機能を用いて行う。

ページめくりを行うときはページめくりモードに設定し、書き込みを行うときは書き込みモードに設定してドラッグ操作を行うと、そのドラッグ操作の運筆データとともに、設定中のモードがデータファイルに出力されるようになっている。設定中のモードがページめくりモードの場合は「page」、書き込みモードの場合は「pen」をデータファイルに出力することでラベルづけを行った。

### 3.2.3 運筆データの単位

運筆データは 10 ミリ秒間隔でサンプリングし、各運筆データの単位は速度 (px/ms)、加速度 (px/ms<sup>2</sup>)、経過時間 (ms)、ドラッグ時間 (ms)、座標 (px)、変位 (px)、ストローク外接矩形幅 (px) とした。筆圧は 0~1 の値を取る。

## 3.3 データ収集

本来は実際に教育現場で学習者用デジタル教科書・教材を使用している児童・生徒を参加者とすべきであるが、今回は初期検討として、社会人 5 名、大学院生 2 名、大学

生 8 名、小学生 1 名の計 16 名から運筆データを収集した。

参加者には、学習者用デジタル教科書・教材が使用される場面、すなわち授業場面に近づけるため、5 分程度の学習動画を見ながら自由に書き込みやページめくりを行うタスクを行ってもらった。

学習者用デジタル教科書・教材の操作には、指を使用する場合とペンを使用する場合があり、入力方法によってデータに差が出ると考えられる。そこで、デバイス付属のペンのみを使用したときと、指のみを使用したときでそれぞれ行ってもらった。

本来参加者とすべき児童・生徒は学習者用デジタル教科書・教材に使い慣れている。参加者をその状態に近づけるため、参加者には事前に、本研究で着目する機能についてデータ取得用デジタル教科書と仕様の近いビューア Lentrance Reader[15]を使用してもらい、操作に慣れてもらった。その後、データ取得用デジタル教科書の操作方法を説明し操作に慣れ次第、タスクを行ってもらった。

収集した運筆データは、欠損値を含むサンプルを除いたサンプル数が 11821、そのうち、書き込みのサンプル数は 11087、ページめくりのサンプル数は 734 である。

## 4. 自動判別手法の設計

### 4.1 判別手法（モデル）の種類

ドラッグ操作開始直後までに得られる運筆データを使用して判別する初動判別モデル、ドラッグ操作終了時までに得られる運筆データを使用して判別する終了時判別モデルを構築する。

また、指使用時とペン使用時の運筆データには差があると予想されることから、指使用時であれば指使用時のデータのみを学習に使用した指専用判別モデルを、ペン使用時であればペン使用時のデータのみを学習に使用したペン専用判別モデルを使用する方が判別精度は高くなると考えられる。これらの専用判別モデルは、入力方法を判別できるデバイスを使用する場合や、入力方法が定まっている場合であれば使用できる。一方、入力方法が定まっていない場合や、入力方法を判別できないデバイスを使用する場合も多く、そのような場合には、指使用時とペン使用時のデータを混合したデータで作成したモデルが適している可能性がある。そこで、二つの判別タイミング（初動、終了時）それぞれに対し、三つの入力方法（指、ペン、混合）を組み合わせた計 6 種類のモデルを構築する。

### 4.2 特徴量の選定

6 種類のモデルそれぞれで使用する特徴量を前章で選定した運筆データから選定する。各判別モデルで使用する特徴量は、収集した運筆データから、目的変数と各特徴量との相関係数を計算し相関が弱い特徴量を除いた後、VIF (Variance Inflation Factor) の値が多重共線性がないとされる 10 未満になるまで VIF 最大の特徴量を一つずつ除い

て選定した。

#### 4.2.1 初動判別モデルの特徴量選定

目的変数と各特徴量との相関係数を求めた結果、弱い相関があるとされる絶対値が 0.3 以上の特徴量は少なく、ほとんどの特徴量は相関が無いと判断される値であった。しかし、特徴量が少なくなりすぎることから、相関係数の絶対値が 0.15 未満のものを除くことで特徴量を選定した。

表 1 に示す通り、初動判別モデル（指）、初動判別モデル（ペン）、初動判別モデル（混合）全てにおいて 7 個の特徴量が選定された。

#### 4.2.2 終了時判別モデルの特徴量選定

目的変数との相関係数の絶対値が 0.3 未満のものを除くことで特徴量を選定した。

表 2 に示す通り、終了時判別モデル（指）では 8 個、終了時判別モデル（ペン）では 13 個、終了時判別モデル（混合）では 8 個の特徴量が選定された。

表 1 初動判別モデルの特徴量

項目	指	ペン	混合
筆圧		○	
前回ドラッグ操作終了時からの経過時間	○	○	○
物理座標系の変位 (Y 成分)		○	○
紙面座標系の変位 (X 成分)		○	○
紙面座標系の変位 (Y 成分)	○		
物理座標系の変位の大きさ	○		
紙面座標系の変位の大きさ		○	○
物理座標系の速度 (X 成分)	○		
物理座標系の加速度	○		○
X 座標	○	○	○
Y 座標	○	○	○

表 2 終了時判別モデルの特徴量

項目	指	ペン	混合
紙面座標系の速度の最大値			○
物理座標系の速度 (X 成分) の最小値	○		○
物理座標系の平均速度 (X 成分)	○	○	○
物理座標系の平均速度	○		
物理座標系の速度 (X 成分) の中央値		○	○
物理座標系の速度の中央値	○		○
紙面座標系の速度の中央値		○	
物理座標系の最終速度 (X 成分)		○	
物理座標系の最終速度 (Y 成分)		○	
物理座標系の最終速度	○	○	○
紙面座標系の平均加速度		○	
物理座標系の加速度 (X 成分) の最大値		○	
紙面座標系の加速度 (X 成分) の最大値	○		
紙面座標系の加速度の最大値	○	○	
物理座標系の加速度の最小値		○	
物理座標系の加速度の中央値		○	○
物理座標系の初加速度		○	
物理座標系のストローク外接矩形幅 (X 軸方向)	○	○	○

#### 4.2.3 選定された特徴量の考察

物理座標系の値と紙面座標系の値は、拡大・縮小をしない限りほぼ同じ値になるため強い相関がある。また、速度は変位をデータ取得間隔で割ったものであり、変位と速度の間には強い相関がある。このような強い相関があり、同じような役割を持つ特徴量は一つに絞られた。

初動判別モデル（ペン）には筆圧が特徴量として選定されているが、初動判別モデル（指）では選定されていない。ペンで入力したときの筆圧の値は 0~1 の幅広い値を取っているが、指で入力したときの筆圧の値はほとんどが 0.98 程度の似たような値になってしまい、初動判別モデル（指）の特徴量として残らなかったと考えられる。

終了時判別モデルにおいて、すべての入力方法で共通して選定されている特徴量に、物理座標系の最終速度がある。書き込みは「とめ」が多く、ページめくりは「はらい」の動作であり、ドラッグ操作終了時付近にその特徴が出ると考え取得したものである。実際に収集したデータを見ると、ページめくりのデータでは平均 2.1、書き込みのデータでは平均 0.2 であり、書き込みとページめくりのドラッグ操作の違いがこの値に出ていると考えられる。

終了時判別モデルにおいて、ストローク外接矩形幅 (X 軸方向) は特徴量として選定されているが、ストローク外接矩形幅 (Y 軸方向) は残っていない。ページめくりのドラッグ操作は横線を引くような動作であり、ドラッグ操作の縦幅に特徴が出ると考えてストローク外接矩形幅 (Y 軸方向) を取得した。しかし、限られたスペースに書くことが多い書き込みよりも、その点気にする必要のないページめくりは書き込みよりも大きくドラッグ操作を行うことが多い。完全に X 軸に平行なドラッグ操作が行えるわけではないため、ドラッグ操作が大きくなるにつれて Y 軸方向の幅も大きくなり、書き込みの幅と同程度になったために、ストローク外接矩形幅 (Y 軸方向) が特徴量として残らなかったと考えられる。ドラッグ操作の大きさに影響されないデータとしてドラッグ角度が考えられる。

#### 4.3 判別モデルの学習手法

今回収集した運筆データは、書き込みのサンプル数が 11087、ページめくりのサンプル数が 734 と不均衡なデータとなっており、学習手法によっては多数派クラスである書き込みに偏って判定するモデルとなる可能性がある。

そこで、各判別モデルの構築には、不均衡データの影響を受けづらい教師あり学習手法の一つである勾配ブースティングを用いる。勾配ブースティングとは、勾配降下法とブースティング、決定木の 3 手法を組み合わせた手法であり、高い精度を得やすい。

### 5. 判別モデルの実装と判別性能評価

#### 5.1 判別モデルの実装

前 3 章で収集した運筆データを使用し、6 種類の判別モ

デルを構築した。

プログラミング言語は Python3.9 を使用し、データセットの整形には Pandas ライブラリを用いた。勾配ブースティングによる判別モデルの構築には、scikit-learn モジュールで提供されている GradientBoostingClassifier を用い、ハイパーパラメータはデフォルト値とした。

データの预处理として、収集したデータのうち、欠損値を含むデータや意図せずドラッグ操作を行ってしまったときのデータは削除した。

## 5.2 判別モデルの判別性能評価

本節では、前節で述べた実装方法で構築した 6 種類の判別モデルの判別性能評価を行う。

### 5.2.1 評価指標

本判別モデルでは、ドラッグ操作を、書き込みをあらわす「pen」とページめくりをあらわす「page」に分類する二値分類を行う。説明のため「pen」を陽性 (Positive) とし「page」を陰性 (Negative) とし、実際のクラスが「pen」で判別結果も「pen」だった場合が TP、実際のクラスが「pen」で判別結果が「page」だった場合が FN、実際のクラスが「page」で判別結果が「pen」だった場合が FP、実際のクラスが「page」で判別結果も「page」だった場合が TN とする。

六つのモデルによるドラッグ操作判別結果の評価では、正しく判別できるかが重要であり、指標としては、実際が書き込みのデータを書き込みであると判断できた割合である [pen]再現率、実際がページめくりのデータを書き込みであると判断できた割合である [page]再現率、書き込みであると判断し実際に書き込みのデータであった割合である [pen]適合率、ページめくりであると判断し実際にページめくりのデータであった割合である [page]適合率があげられる。これらの指標に重要度の差はないため、適合率と再現率のバランスを取った指標である F1 値も用いる。

また、マシューズ相関係数 (MCC : Matthews Correlation Coefficient) は混同行列 TP, FN, FP, TN の全要素を考慮しており、「pen」と「page」の両クラスに対する判別性能が高くない場合 MCC の値は大きくなり、両クラスに対する判別性能をバランス良く評価できる。偏った判定をする判別モデルに対しても適切に評価が可能のため、二値分類モデルの総合的な評価指標として適している。そこで F1 値に合わせて MCC も求める。

正解率、[pen]適合率、[pen]再現率、[pen]F1 値、[page]適合率、[page]再現率、[page]F1 値は 0~1 の値を取り、1 に近いほど良好である。MCC は、-1~1 の値を取り、完璧に分類ができているときに 1、ランダムな分類と同等であるときに 0、完全に真逆に分類しているとき -1 となる。

### 5.2.2 結果

5 分割交差検証を各 5 回行った結果の平均を初動判別モデルは表 3 に、終了時判別モデルは表 4 に示す。

表 3 初動判別モデルの結果

	指	ペン	混合
正解率	0.9722	0.9792	0.9761
[pen]適合率	0.9783	0.9846	0.9814
[pen]再現率	0.9921	0.9935	0.9933
[pen]F1 値	0.9851	0.9891	0.9873
[page]適合率	0.8720	0.8675	0.8771
[page]再現率	0.7117	0.7318	0.7153
[page]F1 値	0.7838	0.7939	0.7880
MCC	0.7736	0.7862	0.7799

表 4 終了時判別モデルの結果

	指	ペン	混合
正解率	0.9890	0.9934	0.9907
[pen]適合率	0.9939	0.9965	0.9948
[pen]再現率	0.9941	0.9966	0.9952
[pen]F1 値	0.9940	0.9965	0.9950
[page]適合率	0.9323	0.9442	0.9339
[page]再現率	0.9298	0.9430	0.9284
[page]F1 値	0.9310	0.9436	0.9311
MCC	0.9251	0.9401	0.9262

初動判別モデルの MCC はどの入力方法でも 0.77 以上であり、判別性能は良好である。終了時判別モデルの MCC はどの入力方法でも 0.92 以上であり、優れた判別性能を有していることがわかる。

初動判別モデルより終了時判別モデルの方が各指標の値が高い理由は、初動判別モデルではドラッグ操作開始直後のデータしか使用できないが、終了時判別モデルでは 1 ストローク全体のデータを使用できるためだと考えられる。

F1 値から、初動判別モデル、終了時判別モデルともに、書き込みのデータに対する判別性能よりも、ページめくりのデータに対する判別性能の方が低く、ページめくりのデータの判別に課題があることがわかる。

初動判別モデル、終了時判別モデルともに、異なる入力方法であっても各指標の値は同程度であり、指かペンか混合かの入力方法による性能差は見られなかった。

これらのことから、デジタル教科書ビューアにおけるドラッグ操作自動判別手法 (モデル) としては、入力方法の制限がなく特徴量も少ない、初動判別モデル (混合) と終了時判別モデル (混合) が最も実用であると考えられる。また、この二つの判別モデルは筆圧に関する運筆データの特徴量として持たないため、筆圧を取得できない端末においても同程度にドラッグ操作を判別できる可能性がある。

## 6. 実機評価

### 6.1 概要

判別モデルの判別性能と判別時間の評価を目的として、リアルタイムに入力されるデータに対する判別性能の客観的評価と、大学生 4 名、大学院生 2 名、社会人 2 名の計 8

名を対象としたアンケートによる主観的評価を行った。

実機評価で使用するモデルは、汎用性の高い初動判別モデル（混合）と終了時判別モデル（混合）とした。前章ではデータ収集で得たデータの約 4/5 を学習データとして構築したモデルを評価したが、ここでは、全サンプルを使用してモデルを構築した。

## 6.2 実機評価環境

実機評価に使用したデバイスは、データ収集時と同じものを使用した。実機評価用 Web アプリケーションとして、データ取得用デジタル教科書に判別モデルを搭載し、判別性能を客観的に評価するための判別結果出力機能、判別精度と判別時間を主観的に評価するための判別結果表示機能を加えた簡易的なビューア（以下、判別結果出力機能付きデジタル教科書）を開発した。

判別結果出力機能により、選択中のモード（実際のクラス）と判別モデルによる判別結果を CSV 形式のデータファイルに出力し、実験後に判別性能を確認できる。判別結果表示機能は、操作画面上に判別結果を表示する機能で、被験者は自分が行ったドラッグ操作が正しく判別されたかを実験時に確認できる。判別精度、判別時間を主観的に評価するために、ドラッグ操作を自動で判別し誤操作を修正する機能を付けたかったが、判別結果を JavaScript プログラムが受け取るまでの時間を高速化できず、この機能を付けるとドラッグ操作をしているときに前回のドラッグ操作に対する修正が行われるなど著しく実験に支障をきたすため、判別結果の表示とした。

判別結果出力機能付きデジタル教科書は、Python3.9, JavaScript, HTML, CSS を使用して開発した。JavaScript プログラムでリアルタイムに取得したデータは Ajax 通信で Python プログラムに送られ、構築しておいたモデルに入力する。得られた判別結果を JavaScript プログラムに渡し、判別結果をデータファイルと操作画面上に出力する。

## 6.3 実機評価方法

被験者に対し、判別タイミングや誤操作修正イメージなど実機評価で使用するモデルについて説明した後、入力方法は制限せずに授業場面を想定しながら自由に書き込みやページめくりをするタスクを行ってもらった。なお、書き込みしたいときは書き込みモードを選択し、ページめくりしたいときはページめくりモードを選択してからドラッグ操作を行ってもらった。

タスク中は、ドラッグ操作 1 回 1 回に対して表示される判別結果を見て、正しく判別されているかと判別結果表示までにかかった時間を確認してもらった。実際の授業場面に近づけて実機評価を行いたかったが、今回作成した判別結果出力機能付きデジタル教科書では、判別結果が操作画面に表示されるまでの時間を高速化できず、短時間で連続してドラッグ操作を行った場合、結果の表示が遅れ、ドラッグ操作一つひとつに対する判別結果を確認することが難

しい。そのため、本実機評価ではその都度判別結果を確認しながらドラッグ操作が行えるよう、自分のペースでドラッグ操作を行ってもらったこととした。

タスク終了後に、被験者には各モデルに対して、「正しく判別が行えていたか」、「判別結果が表示されたタイミングで誤操作が修正されると想定して判別結果が表示されるまでにかかった時間（初動判別モデル（混合）の評価時にはドラッグ操作開始時から判別結果が表示されるまでの時間を、終了時判別モデル（混合）の評価時にはドラッグ操作終了時から判別結果が表示されるまでの時間）が適切であったか」について「とてもそう思う」「そう思う」「どちらでもない」「そう思わない」「全くそう思わない」から選択してもらった五件法で調査し、その理由や誤って判別された場面について自由記述で回答してもらった。

また、被験者に行ってもらったドラッグ操作に対する実際の判別精度検証（以下、実機判別性能評価）を行った。

## 6.4 結果と考察

### 6.4.1 実機判別性能評価

初動判別モデル（混合）と終了時判別モデル（混合）の結果を表 5 に示す。

前章の判別性能評価の結果よりも各指標の値が低い。その理由として、データ収集時とシチュエーションが大きく異なるためにデータの特徴が変わり、判別が困難になった可能性がある。データ収集時とシチュエーションを揃えたり、より多くの人のデータを学習に使用し、さまざまなパターンを学習させたりすることで改善できると考える。

表 5 実機判別性能評価の結果

	初動判別モデル (混合)	終了時判別モデル (混合)
正解率	0.6143	0.9174
[pen]適合率	0.5300	0.9187
[pen]再現率	0.9725	0.9549
[pen]F1 値	0.6861	0.9365
[page]適合率	0.9417	0.9149
[page]再現率	0.3404	0.8515
[page]F1 値	0.5000	0.8821
MCC	0.3842	0.8199

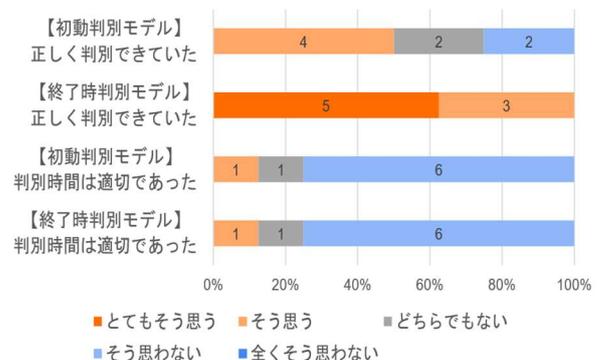


図 2 アンケートで取得した五件法による調査の結果

