LMS を用いたプログラミング授業における機械学習による得点率予測

松尾 龍磨　伊藤 恵

近年, 様々な教育機関でプログラミング教育が進められており, プログラミング学習者が増加している. しかし, 個別での指導体制は難しく, 多人数授業になりがちで, 単位不認定者はますます増加してしまう. さらに, 習熟度に応じた授業が期待されているが, 多人数授業のため, 教員が個々の学生の理解度を把握することは難しい. また, 学生の理解度が低いままで次の演習に進んでしまい, 定期試験の点数が振るわない結果になってしまうという現状にある. そこで, 授業内に収集される学習データを分析することで, 教員が学生のつまずきをいち早く発見し, 単位不認定の可能性のある学生に適切な対応ができるようにすることを目指す. 本研究では, 機械学習を用い試験の得点率の予測を試みる. プログラミング授業における LMS(Learning Management System) 内の過去 6 年分×約 180 人分の学習データを用い, データ間の関係性やパターンを特定し, 目的変数に有益な特徴量を設定することで, 精度の高い予測を目指す. これにより, 教員が理解度の低い学生を早期に発見でき, 単位不認定を防ぎ, 期末試験で点数を取ってもらうように適切な対応ができる.

Recent years, programming education is being promoted at various educational institutions, and programming learners are increasing. However, individual instruction system is difficult and the number of non-credit students increases due to the large number of classes. Furthermore, lessons are expected depending on the degree of understanding. But it is difficult for teachers to grasp the degree of understanding of each student. Also, students proceed to the next exercise with a poor understanding the content, and they can not get the good score for the test. Therefore, by analyzing the training data in the class, we aim that teachers can detect student failures quickly and can provide appropriate support to students who may not be able to earn credits. In this research, we try to predict the score of the test using machine learning. We use the data of the past 6 years × about 180 students in the LMS (Learning Management System) in programming class as the training data. By identifying relationships and patterns between the data, and by setting useful features in the target variables, we aim to attempt highly accurate prediction. Therefore, teachers can detect students with a low level of understanding early and can take appropriate measures for them.

1　はじめに

プログラミング授業は, 大学教育で必須となっていることが多い. しかし, プログラミングについて「難しい」と感じている学生は多い [1]. 大学や高等学校や専門学校で扱われるプログラミング授業は, 授業前半にスライドを用いて説明を受けて, 後半に演習問題を解く形式だったり, 授業中は演習のみでそのコードをレポートとして提出する形式が一般的である. 期限内に提出が必須な課題がある場合だと, その学習内容の理解度が低い学生や学習意欲の低い学生は, 教員の目が届かないところで他の学生の回答を写す可能性が高いと考える. そのため, 教員が LMS 上の成績データだけでは, 学生毎の理解度を把握することは難しい. 布施・岡部（2016）の調査 [2] では, 大学入学時でのプログラミング経験者は全体の 2 割未満とごく僅かである. プログラミング未経験者は, プログラミングの概念を全く把握できていないため, 大学入学し, いきなりプログラミングを学習すると理解が追いつかなく, 授業の学習進度についていけなくなる学生が増えてしまうと考える. それにより, 期末試験で得点が取れなくなり,

Prediction of Score by Machine Learning in Programming Class using LMS
Ryuma Matsuo, 公立はこだて未来大学, Future University Hakodate.
Kei Ito, 公立はこだて未来大学, Future University Hakodate.

単位不認定に繋がってしまう. 本研究では,LMS 上の成績データだけではなく, 提出時の時刻や提出回数などの提出データを用いて, 事前に期末試験の得点率の予測を行う. そして, 教員が単位不認定の可能性がある学生に適切な対応できるようにすることを目指す.

2 関連研究

2.1 学習者のモデルを取り入れたマルチエージェントシミュレーションによる成績予測

芳野ら [3] は, 必修科目であるプログラミング演習の講義を対象に学習履歴をデータマイニングし分析した. 学習者の達成度の評価や将来的な能力の予測などを行う Learning Analytics を利用して, 講義中に実施される 8 回分の理解度確認試験の成績予測を行った. 学習者を記憶率と知識というパラメータを与え, モデル化したエージェントを用いてシミュレーションを行い, 生成されたエージェントの試験結果を学習者と比較し, 近しいエージェントの試験結果から成績を予測する手法をとった. 学習者の得点予測の点で一致している. 本研究では,LMS 上の授業中課題の成績データを用いて, 期末試験の成績予測を行う.

2.2 テキストマイニングによる学習者の特性と理解度の分析

東ら [4] は, 首都圏の私大文系学部で 1 年生向けに開講されている科目「統計入門（必修）」を対象に, 学生毎の授業後の自由記述アンケート, 小テスト,5 段階の理解度アンケートを用いて, 定期試験の得点予測を行なった. 自由記述アンケートについては,「軽い振り返り」「深い振り返り」など 7 つのタグ付けを行い, 定期試験の得点予測における重要度を出した. その結果,「文字数」や「専門用語」が多く,「疑問」が多い学生ほど成績は高い結果となった. 機械学習の予測精度としては,53.3 パーセントとなった. 本研究とは, 教科書クイズ, 演習課題といった授業ごとに行う理解度調査を利用した機械学習という点で一致している. 授業後アンケートは使用しないため,LMS 上の課題提出回数や回答時間を特徴量として追加し, 予測精度の高い機械学習を目指す.

3 対象授業と仮説設定

3.1 授業形態

本研究で対象となるプログラミング授業は, 著者ら所属大学で 2 年次前期に行われる必修科目「情報処理演習 I」であり, 毎年約 180 人が受講している. 使用言語は Java である.Moodle ベースの LMS を利用しており, 事前予習として授業前までに教科書クイズという LMS 上の課題を終える必要がある. 教科書クイズは, ほとんど穴埋め形式になっており, 提出すると正答率が表示される. 授業前までに満点にする必要があり, 何度も提出することが可能である. そして, 実際の授業では, まず, ライブ授業で演習の対象となる学習内容の大まかな説明や教科書クイズの解説を行い, その後, 各々の学内 PC で課題演習を行う. 演習中は, インターネットでの検索や近隣の学生との相談は禁止されていない. しかし, 他人のコードをそのまま写して提出することは禁止されている. 授業内で行う演習課題は,2 題もしくは 3 題ある. 授業後にも提出可能であるが, 遅れた程度に応じて減点される. なお, 教科書クイズと演習課題の内容は, 年度によって多少の違いがある.

3.2 仮説の設定

上記の授業形式の問題点として, 教員の目の届かないところで他の学生のコードを写し, 数行改変して提出するといったチートをする学生がいる点である. これでは, 課題の提出状況では優秀な成績であるが, 学生の理解度は向上しないため, ますます学生の理解度を教員が把握するのは難しくなる. そこで, チートをしている学生と自分で学習を行っている学生を区別できる仮説を以下に記載する.

(a) **教科書クイズの解答時間が長く, 提出回数が多いと, 演習課題の点数が高く, 解答時間が短い**
事前予習である教科書クイズを時間をかけて解答し, かつ何度も提出して満点まで達した学生は, 授業内での演習課題で, いち早く正答できると考えた.

(b) **教科書クイズの解答時間が短く, 提出回数が *1* 回の学生はチートしている可能性が高い**

教科書クイズは一度の提出では, 満点に達することはほぼ困難であるため, 解答時間が短く, 提出回数が 1 回の学生は他の学生の解答を写している可能性が高いと考えた.

(c) **他人と変数名が一緒で, 数行同じコードがある場合はチートしている可能性が高い**
演習課題において, 他の学生のコードを写して提出している場合は, 自分で適当な名前を決められる変数名が同じ場合や, 数行コードが一致する場合が多いと考えた.

4 データ整形

本研究では, 学生別の中間試験と期末試験の得点率が書かれた Excel 形式のデータファイルと LMS 上の学生が提出した教科書クイズと課題演習の提出時間や提出時の得点や提出回数の入ったデータファイルを扱う. これらのファイルを結合し, 一つの CSV 形式のファイルとする. なお, データは全て匿名化されている.

4.1 特徴量選択

仮説に基づき抽出した特徴量を以下に記載する.

- **演習課題が授業時間内に終わった回数**
  演習課題が授業時間内に終わることは, 教科書クイズを他の学生の回答を写さずに学習した可能性が高く, その学習内容の理解度が高いと考えた. 演習課題が授業時間内に終わった回数が多いほど, チートせずに学習していると考えた.
- **各演習課題の点数の平均**
  授業中に公開になる演習課題は, 授業後に満点のものを提出されると 20 パーセントの減点となる. また, 演習課題の途中で授業が終わると, 各々予定がある学生もいるため, 切り上げることも可能である. その際の演習課題の点数は, 最後に提出した点数となる. 取り組んでいない演習課題があった場合, その分の学習内容が理解できないため, 期末試験の得点率に影響を及ぼすと考えた.
- **演習課題の提出回数の平均**
  演習課題が授業時間内に終わる学生や演習課題の得点が高い学生は, チートをしている可能性もあるため, 平均的に提出回数の少ない学生はコードを写して提出する傾向にあると考えた.
- **教科書クイズの提出回数の平均**
  事前課題である教科書クイズを他の学生の回答を写さずに学習しているかを確認する指標になると考えた. しかし, 期末試験の得点率を予測する上で, 直接的には影響度は小さいものだと考えた.
- **中間試験の得点率**
  期末試験の得点率予測の上で中間試験の得点率は, 授業の理解度を数値的に測れるため, 最も重要な特徴量だと考えた.

表 2 は, データ整形後の CSV ファイルを一部抜粋したものである. 今回使用したデータ数は,2 年分 161 人分である. ここでの,「userid」は学生ごとに一意に割り当てられている id のことである. そして,「kad_perfectscore_count」「submitnum_kad_ave」「score_kad_ave」「submitnum_kyou_ave」「midtermtest」「finaltest」はそれぞれ,「演習課題が授業時間内に終わった回数」「演習課題の提出回数の平均」「各演習課題の点数の平均」「教科書クイズの提出回数の平均」「中間試験の得点率」「期末試験の得点率」を示している. 再履修生について, 年度毎に id を同じものにしてしまうと, 提出物が他の学生の倍になってしまい, 特徴量抽出の際に扱いが難しくなってしまうため, 異なる id を割り当てることにした.

表 1 が目的変数である期末試験の得点率に対して, 有益である特徴量を選択するために, それぞれの特徴量と目的変数の相関係数を求めたものである.

そして, 図 1 が各特徴量による予測誤差の二乗平均の減少量に対して, データ点数の重みを掛けた数値を調べることでそれぞれの特徴量の重要度を計算したものである. これについては,scikit-learn の feature_importances_[5] によって算出可能である.

5 機械学習による得点率予測

5.1 手法選択

本研究では, 得点率予測のため, 回帰による機械学習を行う. なお, 教師あり学習であるため, アルゴリズムの種類としては,SupportVectorMachine,RandomForest,RidgeRegression,DecisionTree

特徴量	目的変数との相関係数
演習課題が授業時間内に終わった回数	0.246554
各演習課題の点数の平均	-0.037199
演習課題の提出回数の平均	0.388405
教科書クイズの提出回数の平均	-0.114609
中間試験の得点率	0.637853

表 1　各特徴量と目的変数との相関係数

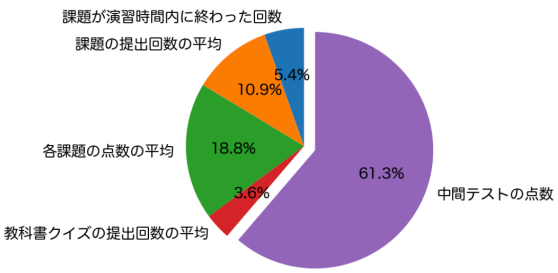

図 1　特徴量の重要度

などが存在する. アルゴリズム選択のために MALSS（MAchine Learning Support System）を利用した.MALSS とは, 機械学習のタスクを容易にする Python モジュール [6] であり, データの前処理，アルゴリズム選択，ハイパーパラメータチューニングといった一連の分析手順を自動化するものである. その結果, 使用するアルゴリズムは,RandomForest とした.RandomForest とはアンサンブル学習の一種であり,Leo Breiman によって 2001 年に提案されたものである [7]. 複数の木を用いて森を構成して識別などを行う機械学習手法であり, 決定木を複数用いることで, 高い予測性能を得る [8].

ハイパーパラメータチューニングについては,scikit-learn にモデルのパラメータをチューニングする仕組みとしてて GridSearchCV [5] が組み込まれている. これを利用し, 最も精度が高いパラメータで RandomForest による機械学習を行うこととした.

5.2　分析結果

特徴量を 5 つで設定した上で,RandomForest による機械学習を行なった. データセットのカラムを特徴量と目的変数に分割し, インデックスを 7 割学習データ,3 割検証データに分割するのが精度を検証する方法として一般的である [9]. しかし, 本研究では, データ数が少ないため,8 割学習データ,2 割検証データとした.scikit-learn の train_test_split 関数を使いランダムにインデックスを分割し,8 割の学習データで教師あり機械学習を行う. そして,2 割の検証データの特徴量を機械学習し, 目的変数の数値を予測する. 結果の数値は 1 点刻みである.

161 人分のデータのうち 128 人分を学習データとし, 残り 33 人分を予測対象としたところ, 予測対象の学生 1 人目については, 予測結果が 75 点で, 実際の値が 82 点であった.2 人目は, 予測結果が 82 点で, 実際の値が 73 点であった. 図 2 は, このようにして予測した全学生 33 人分の期末試験の予測値と実際の値のグラフである. 縦軸が点数, 横軸が何人目の学生かを, 「predict」「true」は, それぞれ「期末試験の予測値」「期末試験の実際の数値」を示している. 予測値と実際の値の誤差は, 全て 20 点未満に抑えることができた. さらに,33 人中 15 人の予測結果が誤差 5 点以下に抑えることができた.

この予測された目的変数の数値の偏差平方和を実際の数値の偏差平方和で割ることで精度が計算できる.

図 2 の分析結果の精度は,80 パーセントとなった. これは一回の実行結果であり,10 回実行すると, 精度の範囲は,42 パーセントから 87 パーセントの幅があり, 平均は 73 パーセントとなった.

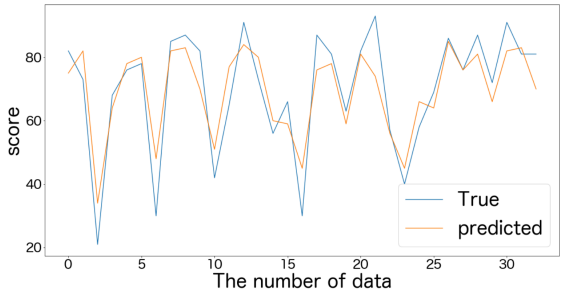

図 2　期末試験の予測値と実際の値

	userid	kad_perfectscore_count	submitnum_kad_ave	score_kad_ave	submitnum_kyou_ave	midtermtest	finaltest
0	183126	23	2.88	9.84	1.36	88	84
1	183127	21	3.36	9.68	1.82	76	93
2	183132	14	2.64	9.04	2.36	58	66
3	183134	21	2.6	9.68	2.27	74	83
4	183143	24	3.04	9.92	2.55	75	74

表 2　データ整形後の CSV ファイルの一部抜粋

6　評価と考察

データセットは全年度をまとめており, 同じ年度だけでの精度はデータ数が少ないこともあり出していない. そのため, 年度による傾向の違いで多少, 精度が下がっている可能性がある.

そして, 精度のばらつきが大きいことから, 過学習（学習データだけに最適化されてしまって汎用性がない状態に陥ること）していると予想される. その他にも, 過学習しているかを判断する指標として, 学習曲線を使用する. 図 3 は 5.2 章の分析結果による学習曲線である. 青色の線は,「学習データでの精度」, 緑色の線は,「交差検証の精度」を示している. 理想的なモデルであれば, サンプル数を大きくしたとき, 学習データに対する予測精度と評価データ (交差検証の精度) に対する予測精度がほぼ同じ値に漸近する. その漸近される値があらかじめ設定した精度よりも高ければ, そのモデルがうまく作られていることを表す [10]. つまり, 学習データの精度と交差検証の精度の差が大きいと過学習していることとなる. 図 3 の場合, 学習データの精度と交差検証の精度の差がかなり大きいので, 過学習している. そのため, 汎化性能（未学習データに対する識別性能）が低い.

また, 仮説 c については, まだ対応する特徴量を導入できていない. 武田ら [11] は, 他人の学生のコードを写して提出を行なった学生をコーディングスタイルから検出する研究を行なっている. インデント, 演算子などに着目し,59 項目の特徴量として抽出し, 盗用の発見に用いるものである. 学生の正確な理解度を確かめるためにも, このような研究を参考にチートを行う学生の抽出を目指す.

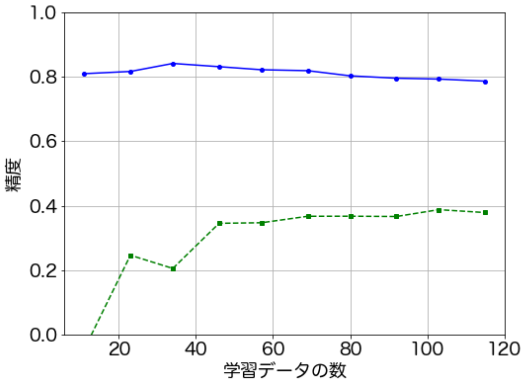

図 3　5.2 章の分析結果による学習曲線

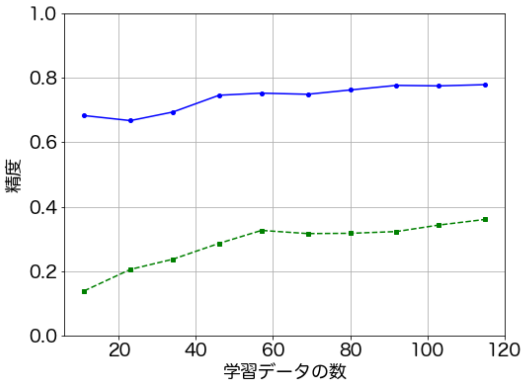

図 4　特徴量削除時の学習曲線

7　おわりに

本稿では, 教員が単位不認定の可能性のある学生をいち早く把握し, 適切な対応をしてもらうことを目的

に,LMS 上の課題提出データから期末試験の得点率予測を行なった.10 回実行した結果の精度として,42 パーセントから 87 パーセントとなった. しかし, 精度のばらつきが大きいほか, 学習曲線から, 過学習していることがわかった. 過学習を防ぐために, データ数を増やし, 汎化性能向上を目指す.

参 考 文 献

[1] 堀越眞理子, 実態調査に基づく一般情報教育としてのプログラミング教育の検討, 筑波学院大学紀要第 14 集,pp87-100(2019)

[2] 布施泉, 岡部成玄：高等教育の一般情報教育におけるプログラミング教育—北海道大学の実践を通して—, 高等教育ジャーナル—高等教育と生涯学習—23, pp53-63(2016)

[3] 芳野洸太, 竹川佳成, 平田圭二, 学習者のモデルを取り入れたマルチエージェントシミュレーションによる成績予測, 第 6 回実践的 IT 教育シンポジウム (rePiT2020) 論文集,2020

[4] 東るみ子, 機械学習による学習者の理解度推定, 教育システム情報学会,pp1-24(2018)

[5] scikit-learn Machine Learning in Python,scikit-learn, https://scikit-learn.org/stable/, (参照 2020-07-19)

[6] MALSS2.3.1,Python Package Index(PyPI), https://pypi.org/project/malss/, (参照 2020-07-24)

[7] L. Breiman: "Random Forests", Machine Learning, 45, 1, pp.5-32 (2001)

[8] 波部斉, ランダムフォレストの基礎と最近の動向, 映像情報メディア学会誌 Vol.70,No.5,pp.788 791(2016)

[9] 秋庭伸也, 杉山阿聖, 寺田学, 見て試してわかる機械学習アルゴリズムの仕組み　機械学習図鑑, 株式会社翔泳社,2019

[10] 学習曲線を使用してモデルの過学習・学習不足を判断, 農学情報科学,https://axa.biopapyrus.jp/machine-learning/model-evaluation/learning-curves.html, 参照 2020-08-08)

[11] 武田隆之, 牛窓朋義, 山内寛己, 門田暁人, 松本健一, コーディングスタイルの特徴量と ソースコード盗用との関係の分析, 情報処理学会研究報告,2009 Information Processing Society of Japan,2009