

ストリームデータによる学習者のプログラミング状況把握

藤原 理也[†] 田口 浩[†] 島田 幸廣^{††} 高田 秀志^{†††} 島川 博光^{†††}

[†]立命館大学大学院理工学研究科 〒525-8577 滋賀県草津市野路東 1-1-1

^{††}株式会社ゴビ 〒600-8813 京都市下京区中堂寺南町 134

^{†††}立命館大学情報理工学部 〒525-8577 滋賀県草津市野路東 1-1-1

E-mail: [†]{masaya,hiro}@de.is.ritsume.ac.jp, ^{††}shimada@go-v.co.jp

^{†††}{htakada,simakawa}@cs.ritsume.ac.jp

あらまし 大学などにおけるプログラミング演習において、教員が個々の学習者の理解度や学習意欲といった取り組み状況を即座に把握することは困難である。そのため、学習者が理解に行き詰まっても、それを演習中に見つけることができず、素早く適切な指導を行うことができない。そこで本論文では、学習者の取り組み状況を表す時系列データから行き詰まりなどの特徴を定量化する手法を提案する。本手法は、ソースコード量やコンパイル回数などの複数の時系列データを視覚的に表現したうえで、そこから直感的に抽出される特徴的な部分を定量化する。抽出したグラフ上の特徴を組み合わせることで、さまざまなプログラミング状況を表現することができる。本手法で定量化した学習中の特徴を用いることで、演習中にその特徴が見られる学習者をリアルタイムで自動把握することができる。

キーワード プログラミング演習, 取り組み状況, ストリームデータ, 特徴の定量化

Grasp of the Learner's Situation in Programming Exercise from Analyzing the Stream Data

Masaya FUJIWARA[†] Hiroshi TAGUCHI[†] Yukihiro SHIMADA^{††}

Hideyuki TAKADA^{†††} and Hiromitsu SHIMAKAWA^{†††}

[†] Graduate School of Science and Engineering, Ritsumeikan University 1-1-1 Noji-Higashi, Kusatsu-shi, Siga, 525-8577 Japan

^{††} GOV Co.,Ltd. 134 Chudojiminami-cho, Shimogyo-ku, Kyoto-shi, Kyoto, 600-8813 Japan

^{†††} Department of Science and Engineering, Ritsumeikan University 1-1-1 Noji-Higashi, Kusatsu-shi, Siga, 525-8577 Japan

E-mail: [†]{masaya,hiro}@de.is.ritsume.ac.jp, ^{††}shimada@go-v.co.jp

^{†††}{htakada,simakawa}@cs.ritsume.ac.jp

Abstract In programming exercise courses, it is difficult for the teacher to know the understanding level of an individual learner. Since no method is provided to grasp the learner attitude for programming, a teacher cannot find a learner who suffers from programming. This paper proposes a method to quantify feature parts of two or more graphing stream data collected from a programming learner. It is possible to look for the learner for whom guidance is necessary by comparison of quantified features of another learner.

Keyword Programming Seminar, Activity Condition, Stream Data, Quantification of Feature

1. はじめに

大学などにおけるプログラミング演習の教育現場では、少数の教員が多数の学習者に指導することが多い。そのため、教員は学習者がどのような学習状況に

あるのかを把握できず、学習者に適切な指導をできないことがある。近年、この問題を解決するために、ネットワークを通じて学習履歴を集めることによって、教員支援を行う研究が行われている[1][2]。しかし、既

存研究では複数の学習履歴に現れる関連情報を見つけることが難しいため、学習者の学習意欲や不正行動などを検知しづらい。

本論文では、時系列で捉えた複数の関連する学習履歴が持つ特徴を定量化することによって、単一の学習履歴の分析では表現できない学習者の状況を表現する手法を提案する。また、定量化した特徴を用いた教員支援について考察を行う。

本手法は、複数の学習履歴をグラフ上で同期表示することにより、学習状況を示す特徴を発見する。また、表現可能なグラフ上の特徴を定性表現としてモデル化しておくことにより、視覚的な操作だけで特徴を抽出する。抽出した特徴は、定性表現モデルに合わせた数式により、定量化される。本手法で定量化した特徴を用いて、プログラミング演習中の学習者を監視することによって、演習中にその特徴が見られる学習者をリアルタイムで自動把握することができる。

2. 学習履歴を用いた教員支援の現状

2.1. プログラミング教育における効果的個人指導

大学などにおけるプログラミング演習では、教員は学習者から質問を受けない限り個人指導を行わないことが多い。そのため、学習に行き詰まったときに質問をしない学習者は学習についていけないことがある。また、学習者が他人のプログラムをコピーするなどの不正を行っても、教員は把握することができないため、不正を行い学習の機会を無駄にしてしまう学習者が現れる。

このような学習者を減らすためには、教員が個々の学習者の学習状況を把握する必要がある。教員が学習者の学習状況を把握することにより、理解が遅れている学習者を把握できるだけでなく、どの学習者に指導を行えば指導効果が高いのかを推測することもできる。たとえば、努力せずに質問ばかりする学習者より、長時間理解に行き詰まっている学習者を優先的に指導することができる。

このように、学習者の状況を把握することで、教員はより適切な指導を行うことが可能となる。しかし、教員が学習者ひとりひとりの学習状況を把握することは非常に困難である。そこで、教員が学習者の学習状況を把握することを支援するシステムが必要である。

2.2. 関連研究

プログラム作成過程における学習者の行動を、獲得・分析することによって、教員支援を行う研究は数多く行われている[1][2][3][4][5][6]。[1]は、コンパイル状況と、エラーの発生状況を記録しておくことによって、クラス全体のエラーの傾向を把握することができる。また、ソースコードの変更箇所も記録しておくことで、学習者の対処行動を把握できる。[2]は、ソースコードの保存時、コンパイル時、プログラムの実行時に学習履歴を記録しておき、履歴を分析してパターンを登録することによって、学習者に自動でアドバイスを

を提供する環境を構築する。[3]は、演習中に学習者が閲覧した資料の時刻、内容、回数を分析することで、学習者の理解状況を把握できる。[4]は、用意されたさまざまな学習者支援機能を実行するときの操作時間間隔に着目し、自動で行き詰まりを検知することができる。[5]は、思考時間とヒントの閲覧時間などの学習過程と、正答率や解答問題数などの学習結果を用いて、学習者の状況の特徴づけることができる。[6]は、他の学習者が閲覧したページをもとにして、学習者に閲覧するページの候補を与えることで、行き詰まりの解消を促すことができる。

これらの既存研究は、学習履歴の分析によって、演習中においてリアルタイムで教員支援を行うことができる。しかし、これらの既存研究ではどの学習者に指導を行えば指導効果が高いのかを推測することはできない。また、専用の学習環境が必要な場合は、学習者に負担をかけてしまうことがある。そこで、専用の学習環境を必要とせず、学習者の学習意欲と行き詰まり状況に着目することによって、指導効果を含めた学習者の状況を把握する環境の構築を目指す。その実現のためには、まず学習者の状況ごとの特徴や傾向を定量化しなければならない。ところが、学習状況ごとの特徴や傾向を定量化する適切な手法は未だ見当たらない。

3. 取り組み状況の推定

3.1. 取り組み状況

教員が学習者に個人指導を行うさい、学習者の状況によって指導効果に変化する。たとえば、学習に行き詰まったときに解決しようと努力している学習者に指導を行えば、高い指導効果が期待できる。また、プログラムをコピーするといった不正を行う学習者には、早く指導するほど指導効果が高いことが予測される。一方、順調に学習を進めている学習者には特に指導の必要がないため、指導効果は低いことが予測される。教員がこれらの学習者の状況を把握することができれば、どの学習者に指導を行うべきかを決定することができる。

本研究では、学習意欲と行き詰まり状況によって表され、教員が学習者に指導を行う基準とすることができる学習者の状況を取り組み状況と呼ぶことにする。教員は取り組み状況を把握することで、適切な指導を行うことができる。しかし、プログラミング演習中に教員が個々の学習者の活動をつぶさに見て回り、取り組み状況を把握することは教員にかかる負担が非常に大きい。そこで、学習者から収集したデータをもとに取り組み状況を表現することで、システムにより自動で取り組み状況を判断することを考える。

著者らは、学習者のソースコードの変化、ソースコードをコンパイルする回数、コンパイル時やプログラムの実行時に出力されるエラーメッセージなどのデータにおける時間に応じた変化に、学習者の取り組み状況が反映されると考える。なぜなら、学習者の学習意欲が高いときは、ソースコードを編集することや、ソ

ソースコードのコンパイルする回数が多いと考えられるからである。そして、学習者が行き詰まっているときは、ソースコードを編集することが少ないことや、エラーメッセージが出続けることが考えられる。また、学習者が他人のプログラムをコピーするときは、ソースコードの行数の急激な増加、コンパイル回数の少なさ、エラー発生数の少なさなどが考えられる。そこで、取り組み状況を示すと考えられる

- ・ ソースコードの行数
- ・ コメントの行数
- ・ コンパイルの回数
- ・ コンパイル時のエラー数
- ・ プログラムの実行回数

の時系列データを、取り組み状況を示すデータ項目として、本研究では扱う。そして、これらの値の変化、またはその組み合わせから取り組み状況を推定する。

3.2. 実験

取り組み状況データから取り組み状況を推定できる可能性を検証するための実験を行った。実験概要図を図1に示す。実験は、のべ7人の学習者にC言語プログラミングの穴埋め問題を60分間解いてもらった。そのさい、ソースコードエディタであるMeadowと、コンパイルを行うためのCygwinの機能を用いて、1分間隔で前述の5種類の取り組み状況を示すデータ項目値と、ソースコードのログ、エラーメッセージを記録した。

図1に示すように、学習者から記録した時系列の取り組み状況データ項目の値の推移を、プログラミング終了後に時間軸を合わせてグラフ表示した。同時に、

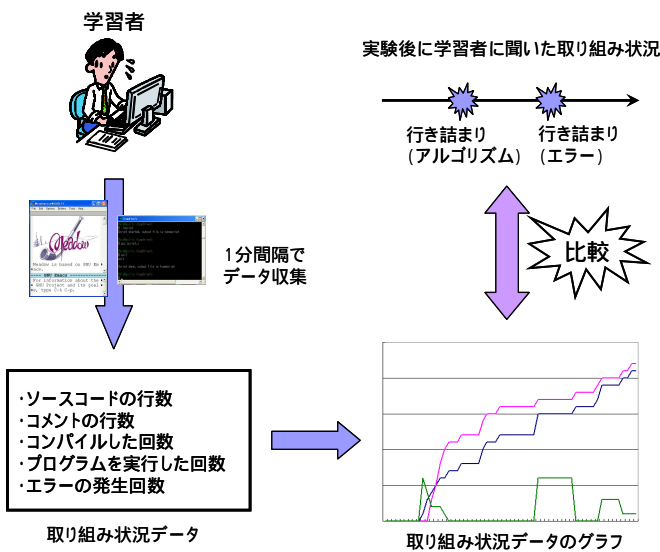


図1：実験概要図

実験後にソースコードのログ、エラーメッセージのログをもとに、学習者から実際の取り組み状況を聞いた。そして、グラフ化した取り組み状況データ項目の値の推移から推測できる取り組み状況と、実験後に学習者から聞いた取り組み状況を比較して、取り組み状況が、収集したデータ項目から推定できるかを検証した。

3.3. 推定可能性

7ケースの実験データを、数人のプログラミング経験者が目視により分析した結果、取り組み状況データが学習者の取り組み状況を示す可能性が高いことを確認できた。特に、コンパイルの回数とエラーの発生回数の2種類のデータ項目値の組み合わせが非常に有効であることが判明した。単一のデータ項目値を分析するだけでは分からない学習者の行き詰まり状況も推測できることがある。グラフ化した取り組み状況の分析にかかる時間は、60分かけて収集したデータに対しおよそ1分ほどであった。取り組み状況データ項目値の推移をグラフ化することによって分析にかかる時間が短く、グラフ上に現れる特徴部分は取り組み状況を示す可能性が高いため、グラフの特徴を定量化する適切な手法があれば、実用性は高いと考えられる。

図2に実験データの1例を示す。図2のグラフの点線で囲まれた部分では、学習者がコンパイルを繰り返し、コンパイル時にエラーはないが、ソースコードの行数やコメントの行数はほとんど変化していないことが分かる。これは、プログラムのアルゴリズムに問題があり、うまく動かずに悩んでいるからであると推測できる。

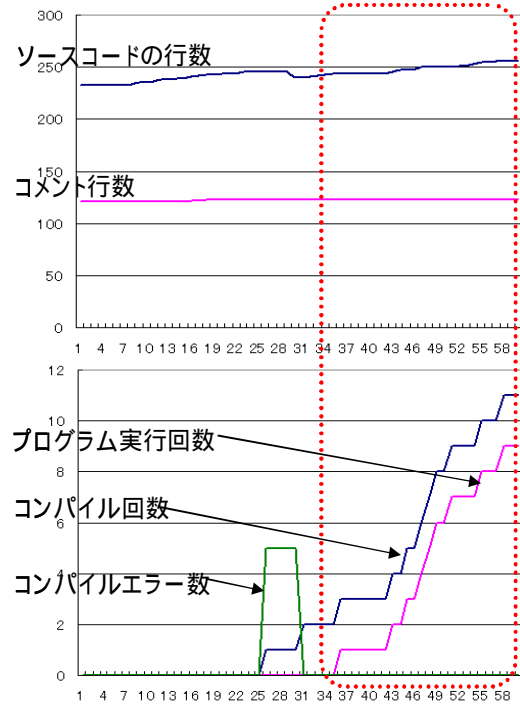


図2：取り組み状況データ

4. 取り組み状況の表現

4.1. グラフ化した取り組み状況データの特徴表現

本研究では、取り組み状況をグラフ化することによって把握可能な学習者の取り組み状況を、視覚的な操作によって数式の組み合わせに定量化する手法を提案する。複雑な計算が必要なく、直感的に定量化を行うことができるため、プログラミング研究者だけでなく、プログラミング学習の教員も利用することができる。取り組み状況ごとの特徴を定量化するまでの流れを図3に示す。

図3に示すように、学習者から収集したそれぞれの取り組み状況を示すデータ項目値を、時間軸を合わせてグラフ表示する。また、学習者から聞いた情報や、ソースコードなどを基にして正確な学習者の取り組み状況を把握する。学習者が行き詰まりなどの特定の取り組み状況にあるとき、取り組み状況データのグラフにどのような特徴が現れるのかを分析する。グラフに特徴を発見した場合、グラフの特徴を数式に変換する定性表現モデルを当てはめることによって、特定の取り組み状況に現れるデータ項目値の推移の特徴を数式として抽出する[7]。このとき、用意されている定性表現モデルを用いて定量化を行うため、定量化のさいに複雑な計算を行う必要がなく、グラフ上の特徴を視覚的な操作だけで表現することが可能である。見つけ出した特徴のすべてに定性表現モデルを当てはめると、特徴が数式に変換される。本手法により、学習者のさまざまな取り組み状況を、視覚的な操作だけで定量モデルとして表現することができる。

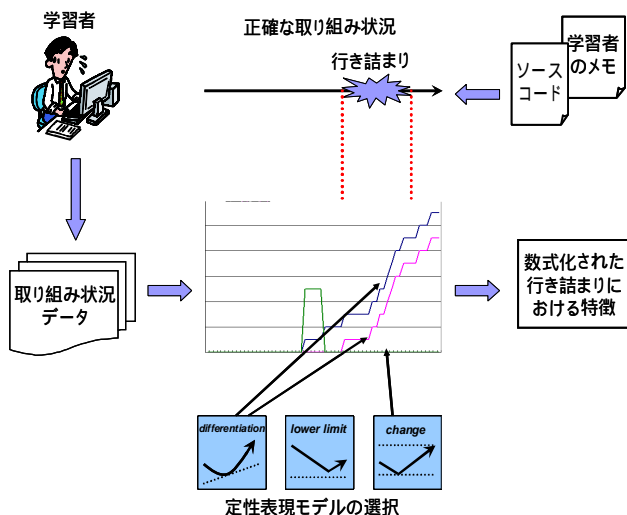


図3：取り組み状況の定量化

4.2. 取り組み状況データ項目の選択

学習者の取り組み状況を正確に表現するには、表現したい取り組み状況に応じて多くのデータから適切な

取り組み状況データを選択しなければならない。著者が行った取り組み状況データの有効性を検証する実験では、学習意欲は高いが長時間行き詰まっている学習者や、プログラムのコピーなどの不正を行う学習者などを対象として、取り組み状況を表現することを目的としている。そこで、ソースコード量の変化と、コンパイルなどを行う端末の操作に注目して5つのデータ項目を選択した。しかし、5つのデータ項目以外にも、取り組み状況を表すと考えられるデータ項目は数多く存在する。たとえば、特定のエラーメッセージの出現数や、ソースコード中の関数の出現数を取り入れることによって、学習者がどのようなエラーに行き詰まっているのかを表現できる可能性がある。また、ソースコード中の予約語の数や、関数の数を取り入れることによって、ソースコードの書き方について表現できる可能性がある。以上のように、本手法ではさまざまなデータの分析が可能であり、表現できる取り組み状況は数多く存在する。そのため、本手法を用いて取り組み状況の定量化を行うさい、利用者の用途に応じた使い方が可能である。たとえば、プログラミング学習の教員が使用する場合、教員の指導方針に応じて不正を行う学習者の特徴を重点的に分析したり、学習意欲の低い学習者の特徴を分析したりするといったことが可能である。本手法を用いるさい、取り組み状況に応じて必要な取り組み状況データを加えることによって、学習意欲と行き詰まり状況以外の学習者の状況を定量化できることも考えられる。

4.3. 定性表現モデルを用いた特徴の検知と定量化

取り組み状況の多くは、単一の取り組み状況データから推測することは困難である。しかし、複数の学習者データを同時に分析し、関連を見つけることによって、推測できる可能性は大きく上昇する。そこで本手法では、時系列データとして捉えた複数の取り組み状況を示すデータ項目値の推移を、時間軸を合わせて同時にグラフ表示することで、データ間の関連に重点をおいた取り組み状況の視覚化を行い、特徴を検知する。

グラフ化した取り組み状況データに現れる特徴を検知した後、検知した特徴を定量化する手法を図4に示す。図4に示すように本手法では、グラフ表示された取り組み状況データの特徴を定量化するために、定性表現モデルを用いる。定性表現モデルとは、データの上昇や上限といったグラフ上に現れる特徴をモデル化したものである。つまり、グラフ上の特徴はすべて定性表現モデルの組み合わせで表現することができる。また、定性表現モデルは、モデルごとにその特徴に応じた数式への変換式を持っているため、定性表現と定量表現を同時に行うことができる。本手法で用いる定性表現モデルは、データ値の増減、データ値の変化量、データ値の上限、データ値の下限、データの微分値の増減の5つである。これら5つの定性表現によって、グラフ上の特徴を表現することができる。グラフ上に特徴を発見したとき、適切な定性表現モデルを選択し、特徴部分に当てはめることで、数式に変換される。このとき、モデルを当てはめたあとに上限値、下限値などの詳細な値を設定することで、より正確に特徴を表現できる。たとえば、モデルを当てはめたときの時間

幅が30分であっても、その後の設定によって時間幅を30分以上に設定したり、30分未満に設定したりすることができる。本手法によって、複雑な計算をすることなく、直感的な操作だけで数式に定量化することができる。

グラフ上における複数の特徴が1つの取り組み状況を示す場合、必要なだけ定性表現モデルを当てはめる。すべての特徴を定性表現に当てはめたとき、最も大きな時間幅を基準として、1つの取り組み状況に定量化される。

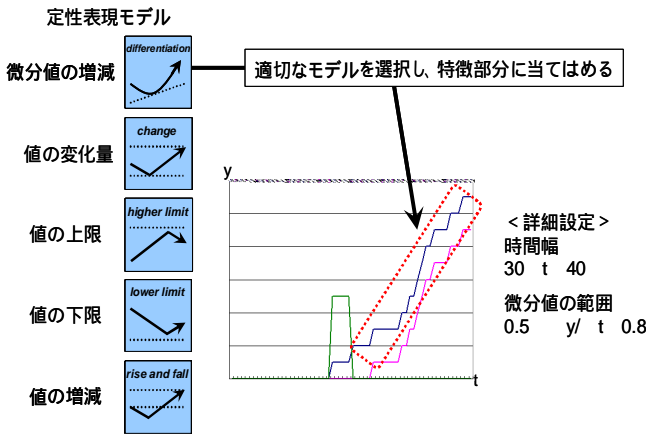


図4：定性表現を用いた定量化

4.4. 定量化された取り組み状況と応用例

3.2で行った実験のデータをもとに、行き詰まり状況の定量化例を示す。本状況において、グラフ上に4つの特徴を発見した。本状況の特徴を定量化した結果を以下に示す。

- ・ソースコードの行数があまり増えていない：
値の増減 $y(\text{line}) \ 10 \{ \ t(\text{min})=25 \}$
- ・コンパイルの回数が増えている：
微分値の増減 $y(\text{count})/ \ t(\text{min}) \ 0.36 \{ \ t(\text{min})=25 \}$
- ・プログラムの実行回数が増えている：
微分値の増減 $y(\text{count})/ \ t(\text{min}) \ 0.36 \ { \ t(\text{min})=25 \}$
- ・コンパイル時のエラー数が0：
値の変化量 $y(\text{count})=0 \ { \ t(\text{min})=25 \}$
- ・取り組み状況の時間幅：
 $t(\text{min})=25$

発見したそれぞれの特徴について、適した定性表現モデルを当てはめることで、特定の時間幅における状況を定量化することができる。上記のような学習者から収集したデータをもとに定量化した状況を蓄積していくことによって、さまざまな取り組み状況を自動把握できる教員支援システムの構築が可能となる。

教員支援システムのイメージを図5に示す。図5はプログラミング演習中における学習者、教員、教員支援システムの関係を示している。演習中に一定時間ご

とにそれぞれの学習者から取り組み状況データを収集する。そして、収集された取り組み状況データと、予め定量化された取り組み状況を比較して、一致するかを判断する。そして、一致した場合、定量化された取り組み状況をもとに教員に学習者の取り組み状況を伝える。本システムによって、教員はリアルタイムに学習者の取り組み状況を知ることができる。

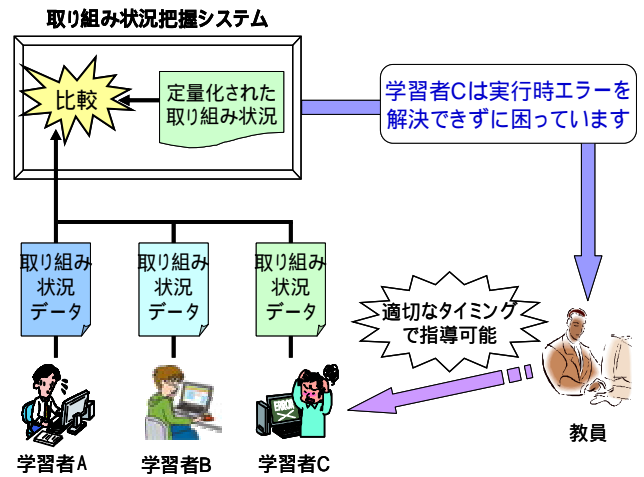


図5：取り組み状況自動把握システム

5. 関連研究との比較

本稿では、学習者の取り組み状況を定量化する手法と、定量化された特徴を用いた教員支援方法までを提案した。本研究を評価するために、関連研究の中で、分析する学習履歴が類似している[1]と[2]について比較を行う。本研究と比較した結果を表1に示す。

まず、本研究の利点として専用の学習環境を構築する必要がないことが挙げられる。専用の学習環境において、学習者の使用するエディタやコンパイラを限定したり、参考にするページを指定限定したりすると、学習者にかかる負担が大きい。本研究では学習者の学習する環境を変える必要がないため学習者への負担はないが、研究者か教員が予め学習状況の定量化を行わなければ教員支援を行うことはできない。また、専用の学習環境を構築しないため、学習者への自動支援を行うことはできない。データ解析における本研究の特徴としては、複数のデータの関連情報を分析できることが挙げられる。それゆえ、それぞれのデータを個別に分析するだけでは分からない学習者の状況を把握することができる。把握可能な学習者の状況は、行き詰まり状況だけでなく、学習意欲や他人のプログラムをコピーするといった不正状況も含まれる。

関連研究と比較した結果、本研究は複数のデータ間の関連情報を分析できるという点が最大の特徴であるといえる。本研究では、既存研究において把握できなかった特徴を把握できる。しかし、有効な特徴を定量化しなければ教員支援することができない。

表 1 : 既存研究との比較

	[1]	[2]	本研究
専用の学習環境が不要		×	
分析済みのデータが不要		×	×
データ分析にかかる時間	-	×	×
データ間の関連情報の解析	-	×	
学習者への自動支援	×		×
教員にかかる負担	×		
行き詰まり状況の把握	×		
学習者の対処行動の把握		×	×
学習意欲の把握	×		
クラス全体の傾向の把握			×

6. おわりに

本論文では，時系列で捉えた複数の学習履歴に現れる特徴を定量化することによって，学習者の状況を表現する手法を提案した．本手法により，学習者ごとの指導効果の高さや，学習意欲，不正状況といった，これまで発見することが困難であった学習者の状況を定量化できる．今後は学習履歴から定量化した特徴を，他の学習者に適用できるかの検証を重ね、学習状況の把握精度についての検討を行う予定である。

文 献

- [1] 諏訪正則，倉澤邦美，鈴木恵介，森本康彦，横山節雄，佐々木整，宮寺庸造，“プログラミング教育における学習履歴取得システムの開発”，第2回情報科学技術フォーラム(FIT2003)講演論文集，pp.583-584，2003
- [2] 長慎也，筧捷彦，“proGrep – プログラミング学習履歴検索システム”，情報処理学会論文誌，Vol.71，No.4，pp.29-36，2005
- [3] 山川修，田中武之，菊澤正裕，“LMS を使った学習プロセスの分析と評価”，第1回 WebCT 研究会予稿集，pp.19-24，2003
- [4] 中村喜宏，赤松則男，桑原恒夫，玉城幹介，“操作時間間隔の変動に着目した CAI 学習の行き詰まり検知方法”，電子情報通信学会論文誌，Vol.J85-D-1，No.1，pp.79-90，2002
- [5] 三田泰正，藤岡健史，萩野哲男，高田秀志，上林彌彦，“学習履歴を利用した動的な問題提示を行う学習支援システムの提案”，電子情報通信学会第15回データ工学ワークショップ(DEWS2004)，2004
- [6] 上田真由美，梶田将司，間瀬健二，“学習操作履歴を用いた利用者コンテキストウェアな CMS ツールに関する検討”，電子情報通信学会第17回データ工学ワークショップ(DEWS2006)，2006
- [7] H.Shimakawa, H.Yamahara, Y.Imayama, M.Ushijima, S.Azuma, Pattern Refinement with Model Data Fusion to Predict Exchange Rate Movement, in A>Abraham, et.al, ed. , Design and Application of