

プログラミングに対する自信と諸要因の因果関係の
ベイジアンネットワークを用いた検証

宮西 和機 齋藤 健司

北海道情報大学

Verification of causal relation of self-confidence in computer
programming and various factors using Bayesian network

Kazuki MIYANISHI and Kenji SAITO

Hokkaido Information University

平成31年 3 月

北海道情報大学紀要 第30巻 第 2 号別刷

〈論 文〉

プログラミングに対する自信と諸要因の因果関係のベイジアン
ネットワークを用いた検証

宮西 和機* 齋藤 健司†

**Verification of causal relation of self-confidence in computer
programming and various factors using Bayesian network**

KAZUKI MIYANISHI* KENJI SAITO†

要旨

プログラミングに対する自信に影響する要因を調査するためにアンケートの実施とベイジアンネットワークを用いた分析を行った。その結果、プログラムの自作経験がプログラミングに対する自信に影響を与えることがわかった。また、事象の因果関係を分析するにあたり、ベイジアンネットワークの有用性を考察した。

Abstract

In this paper, we conducted a questionnaire to investigate factors that affect self-confidence in computer programming and showed analysis results using Bayesian network. As a result, it was suggested that the experience of creating a self-made computer program affects self-confidence in computer programming. We also considered the usefulness of the Bayesian network in analyzing causal relationships of events.

キーワード

プログラミング学習 (Learning programming), ベイジアンネットワーク (Bayesian Network)

* 北海道情報大学情報メディア学部情報メディア学科助教, Assistant Professor Faculty of Information Media, HIU

† 北海道情報大学情報メディア学部情報メディア学科准教授, Associate Professor Faculty of Information Media, HIU

1. はじめに

プログラミングを学ぶ者にとって、その学習の効果や効率を高めることは重要な事柄である。どのような学習スタイルで取り組めば効果的にプログラミングを学べるのか、プログラミング以外の知識はプログラミングの学習効率に影響を与えるのか、というような、プログラミング学習をより良いものとするための方法論に関して、学習者は関心を寄せるものと思われる。

プログラミングの学習スタイルには、サンプルプログラムを模倣して入力、実行し、その結果を確認するという「写経型学習過程」(岡本他, 2013) と呼ばれる手段や、プログラミング言語の仕様の概念の理解を優先して学ぶ方法などが考えられる。

プログラミング学習における初学者のつまずきに関しては、学習を困難にする要因の類型化と学習方略が提案されている(岡本・喜多, 2014)。一方で、プログラミングの学習を促進させる要因については一般的にあまり知られていない。

プログラミング学習が促進された結果として、学習者はプログラミングに対して自信を持つものと考えられる。そこで、プログラミングに対して自信を持つ者と持たない者の間に、学習スタイルの違いや、プログラミング以外の知識の差、その他何らかの違いに対して、プログラミングに対する自信とその要因を推察することができれば、効果的な、または効率的なプログラミング学習の方法論に関する足がかりを得ることができると考え、そのためのデータ収集と推論を試みた。

集めたデータに対して単純な統計処理を行うのみの場合、何らかの事柄とプログラミングに対する自信の関係について相関を調べることができるが、その因果関係についてはその関係を見出すことは難しい。そこで、確率推論を用いたベイジアンネットワークを用いて確率的に因果関係を見出すことを試みた。

1-1 ベイジアンネットワーク

ベイジアンネットワークとは、確率変数を有向グラフのノードとし、ノード間の確率的な関係をリンクを用いて表現することによって得られる確率モデルである(Pearl, 1988; Heckerman, 1995)。図1はベイジアンネットワークの

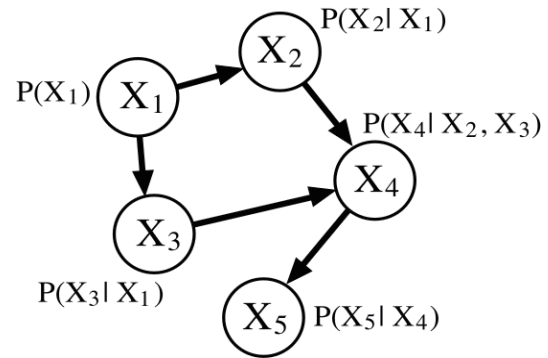


図1 ベイジアンネットワークの例

一例を図示した物である。この図の中には X_1 から X_5 までの確率変数が含まれており、確率的に関係のある確率変数間が有向線分によって接続されており、確率的な関係は条件付き確率によりモデル化される。この図では X_1 から X_2 に向って矢印が引かれているが、これにより X_1 と X_2 の確率変数は独立ではなく従属性があることを示している。 X_2 の確率変数の図には $P(X_2|X_1)$ という条件付き確率分が書き込まれているが、これは X_2 の確率分布が X_1 の値によって変化することを意味しており、この分布を調べることで従属性の強さや正の相関があるのか負の相関があるのかを判断できる。

ベイジアンネットワークは多数の事象が互いに影響を及ぼし合っている現象をモデル化するのに適しており、医療診断、故障診断、自律移動システムのナビゲーション、ユーザーの心理モデル構築など多くの応用例がある(Russell and Norvig, 1997; 本村, 2003)。

ベイジアンネットワークを用いた確率モデルを構築するためには、項目間の接続の決定と、各項目が表わす確率変数の条件付き確率分布の決定が必要である。これらの作業を、その分野の知識のある専門家の助けを借りて人手で行う場合もあるが、多くの場合は大量のデータを用いて機械学習の技術を使うことで自動的に確率モデルを構築する。本研究では学生のアンケートへの回答のデータを用いてベイジアンネットワークを機械学習させることになるが、これによりアンケートのどの質問項目間に関連があり、どの項目間に関連が無いのかを自動的に図示することが可能となる。

データからの確率モデルの構築や推定分野では、過剰適合やオーバーフィッティング、または過学習と呼ばれる問題があるが、ベイジアンネットワークの学習においては各種情報量基準

を用いた学習アルゴリズムを採用することが多く、これらの問題に対する一定の理論的裏付けがなされている。本研究ではMDL(Minimum Description Length, 最小記述長) (韓・小林, 1994) と呼ばれる情報理論に基く情報量基準を採用した。

MDL原理は、その確率モデルを記述するための情報量と、その確率モデルを用いて学習データを圧縮した時の情報量の合計が最小となる確率モデルを良いモデルとして数値化する。これは物事を説明する複数の方法がある時は最も単純で明解な方法を採用すべきであるという「オッカムの剃刀」という哲学を過学習などの問題に適応した物と考えることができ、実際にこの方法を採用することにより、学習データへの適応度と確率モデルの複雑さとのバランスが自動的に調整され、学習に使用していないテストデータでの適応率が高くなることが示されている。

2. 目的

本研究の目的は、良いプログラミング学習の方法論の考察の足がかりとなるような、プログラミング学習者のプログラミングに対する自信に良い影響を与える要因を見つけ出すことである。

3. 方法

北海道情報大学の学生に対してアンケートを実施し、そこで集められたデータをもとに、独自に開発したベイジアンネットワークを用いた推論ソフトウェアである「bNet」を用いて推論を行う。

3-1 アンケートの内容

アンケートに含めた項目は、①学年、②大学の科目、または独学でプログラミングを学んだことがある、または現在学んでいるか、③プログラミングに自信があるか、④どのようなスタイルでプログラミングを学んでいるか、⑤プログラミングの勉強にどのくらい時間をかけているか、⑥大学の授業とは関係のないプログラムを自分で作ったことがあるか、⑦これまで受けてきたプログラミング系の授業で楽しいと感じたことがあるか、⑧コンピュータが動く仕組みについて理解しているか、⑨数学は得意か、⑩英語は得意か、である。

②大学の科目、または独学でプログラミングを学んだことがある、または現在学んでいるか、の項目は「大学の科目と独学の両方で学んでいる」「大学の科目でのみ学んでいる」「独学でのみ学んでいる」「プログラミングを学んでいない」の中から1つ選択する方式とした。

③プログラミングに自信があるか、の項目と、⑧コンピュータが動く仕組みについて理解しているか、の項目、⑨数学は得意か、の項目、⑩英語は得意か、の項目については5段階評価での回答とした。

④どのようなスタイルでプログラミングを学んでいるか、については「サンプルを書き写して動作させるスタイル」と「プログラミング言語の仕様の概念を理解していくスタイル」を複数回答ありでの回答とした。

⑤プログラミングの勉強にどのくらい時間をかけているか、の項目は「まったくしていない」「大学の授業でやる分だけ」「大学の授業の予習復習もやる」「授業とは関係のないこともやる」の中から1つ選択する方式とした。

3-2 ベイジアンネットワークを用いた推論ソフトウェア「bNet」

今回、ベイジアンネットワークを用いた分析をするにあたり独自のソフトウェア「bNet」を使用した。このソフトウェアは考慮すべき項目と、その項目に対応する確率変数が従うべき大量のデータを与えることにより、ネットワークの構造を自動的に学習して最適なベイジアンネットワークを出力する。

図2は、このソフトウェアのユーザーインターフェースである。このインターフェースを用いることで項目の追加、学習データの読み込みや学習の開始などをマウスやコマンドの入力により実行することができる。学習後のベイジアンネットワークを分析するための機能も実装した。図2はグラフィカルなユーザーインターフェースであるが、コマンドラインのみを持つユーザーインターフェース(CUI)もあり、グラフィカルユーザーインターフェースを持たないサーバーで実行させることも可能である。

3-3 ベイジアンネットワークの学習アルゴリズム

前述したように、bNetではデータからベイジアンネットワークを機械学習により構築す

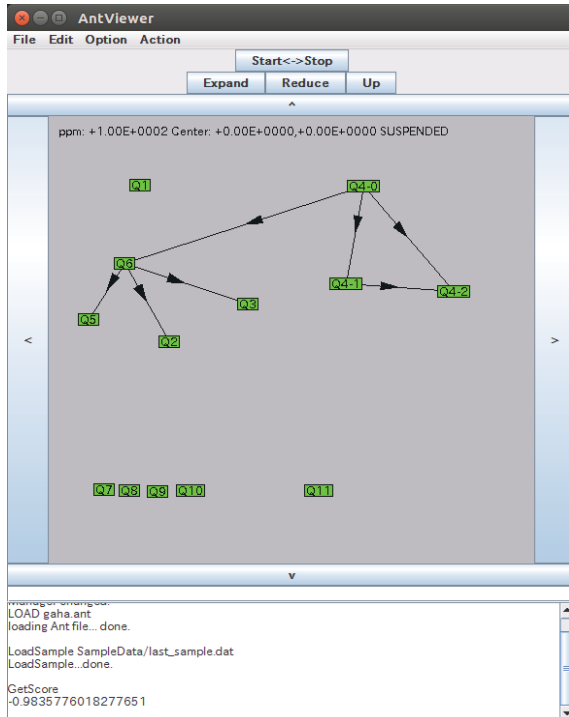


図2 bNet のユーザーインターフェース

るが、本研究で採用したアルゴリズムについて説明する。

まず、学習に使用するデータの内容について説明する。データの内容は分析したいと考えている項目に対応する標本の列である。現在のところ bNet は離散的なデータのみに対応しているので、実際のデータは複数の項目に対応する整数の列となる。今回は学生のアンケート結果を学習させるために、アンケートの1つの質問が1つのノードに対応し、質問における各選択肢を整数に対応付けすることでデータを作成している。複数の選択肢を選択できる質問の場合は、それぞれの選択肢を独立した2値の確率変数とした。自由解答欄を含む選択肢については、入力項目は無視して自由解答欄を選択したことのみデータに反映させている。これらのデータは Google フォームの出力する CSV ファイルから生成することができるが、この生成処理をサポートするプログラムも作成した。

第1-1節にて説明したように、ベイジアンネットワークを用いた確率モデルを構築するためには、(1) 各ノードに対応する確率変数の条件付き確率分布の決定と、(2) ノード間の接続の決定が必要である。実際には(2)を決定してから(1)を決定する必要があるがまず(1)を説明し、次に(2)を説明する。

表1 $P(X_{Q1})$ の確率表

X_{Q1}	$P(X_{Q1})$
1	0.10
2	0.48
3	0.23
4	0.19

3-3-1 確率変数の条件付き確率分布の決定

アンケートの各回答内容を離散型の確率変数 $X_i (i = 1, \dots, M)$ で表すことにする。ここで、 i は質問番号(ノードの通し番号)、 M は質問の総数をほぼ表わしている。本研究では1-1節で述べたベイジアンネットワークモデルの考え方にに基づき、ノード X_i に関する各値の発生確率 $P(X_i)$ が X_i とは異なるノード $X_{c_j} (j = 1, \dots, l)$ の発生確率の影響を潜在的に受け取ると仮定し、この状況を調べるために条件付き確率 $P(X_i | X_{c_1}, \dots, X_{c_l})$ に着目する。ここで、 l は $P(X_i)$ に影響を与えるノードの総数、 c_j はその具体的なノードの番号を意味し、これらの値は未知とする。また $l = 0$ の場合には $P(X_i | X_{c_1}, \dots, X_{c_l}) = P(X_i)$ となるものとする。

本研究では $P(X_i | X_k) (i \neq k)$, あるいは $P(X_i)$ の頻度確率をアンケートの回答結果に基づいて集計し、その値を発生確率にもつ確率分布を確率表 (PT: Probability Table) もしくは条件付き確率表 (CPT: Conditional Probability Table) として定義する。アンケートの最初の質問項目である回答者の学年について、この確率表の例を挙げることにする。この学年の質問事項への回答は後に示すように他の質問事項への回答とは独立しており、別の質問事項の影響を受けることは無かった。このような場合、アンケートに回答した学生の学年を表す確率変数を X_{Q1} とし、この確率変数の分布を表1のように表すことができる。

X_{Q1} の確率表は1年から4年までのそれぞれの事象の確率そのものであるが、離散的な確率分布を表現する確率モデルであると見なすことができる。表中の数値は確率であるために、全てを合計した時に1にならなければならないという制約がある。この制約によって、表1の確率表に基づく確率分布に関する自由度は3となる。この自由度は後程説明する MDL 値を計算するときに必要な数値となる。

次に条件付き確率表を例をあげて説明する。アンケートの中に、「大学の授業とは関係のな

表 2 $P(X_{Q6}|X_{Q4-0})$ の条件付き確率表

X_{Q6}	X_{Q4-0}	$P(X_{Q6} X_{Q4-0})$
T	T	0.32
F	T	0.68
T	F	0.62
F	F	0.38

「プログラムを自分で作ったことがあるか？」という質問 (Q6) がある。また、どのようなスタイルでプログラミングを学習しているかを問う質問 (Q4) があり、複数の選択から自分にあてはまる物を複数選択する形で回答する。その選択肢の中に「サンプルを書き写して動作させるスタイル」という選択肢 (Q4-0) がある。どちらの場合も Yes/No や、選択する/しないの 2 値をとるデータで、ここではそれを真 (T) と偽 (F) で表すことにする。質問 (Q6) を確率変数 X_{Q6} 、選択肢 (Q4-0) を確率変数 X_{Q4-0} で表した時に、 X_{Q6} の分布を条件付き確率表にした物が表 2 である。

この表からは様々なことを読み取ることができる。条件に対応する X_{Q4-0} が T であるか F であるかによって X_{Q6} の確率分布が変化している。 $X_{Q4-0} = T$ の時は X_{Q6} は F になりやすく、 $X_{Q4-0} = F$ の時は X_{Q6} は T になりやすい。つまり X_{Q4-0} と X_{Q6} の間には弱い負の相関があると言える。表 2 の数値は確率を表す数値であるため $P(X_{Q6} = T|X_{Q4-0} = T) + P(X_{Q6} = F|X_{Q4-0} = T) = 1$ と $P(X_{Q6} = T|X_{Q4-0} = F) + P(X_{Q6} = F|X_{Q4-0} = F) = 1$ の関係が成り立たなければならない。よって、表 2 には 4 つの数値が存在するが、自由度は 2 となる。

これら (条件付き) 確率分布関数の値を求める計算は、欠損などの問題や、隠れた確率変数などの難しい状況は存在しないので「(該当するデータの数) ÷ (全体のデータの数)」という単純な計算であり、最尤推定で確率モデルのパラメータを決定する場合と同等の性質を持つ。

3-3-2 ノード間の接続の決定

次に、どのノードがどのノードから影響を受けておりどのように結合させてネットワークを形成するのかを定めるアルゴリズムについて説明する。これを考えるにあたり、どのようなネットワークが良いネットワークで、どのようなネットワークが悪いネットワークかを判断するための指標が必要である。良いネットワークとは、(1) ノード間の関係性を現実に即

した形で表しており、(2) そのネットワークを使うことで学習データを良く説明することができ、(3) 未知のテストデータに関しても良い精度で予測・推論ができる、ことを意味している。ベイジアンネットワークは全体として見た時に、全てのノードに対応する確率変数の同時確率分布を表す確率モデルと見なされるが、「学習データを良く説明できる」とは学習データを前提とした時の確率モデルの尤度により数値化できる。ただ、一般的にこの尤度を大きくすることだけを目標とすると、過剰適合が起り未知のテストデータを予測・推論する性能が低くなってしまう。この問題を解決するためにベイジアンネットワークの学習においては様々な情報量基準をネットワークの良し悪しの基準として採用する機械学習が採用されることが多い。本研究ではこの情報量基準として MDL (Minimum Description Length, 最小記述長) を採用している。これは第 1-1 節でも説明した「オッカムの剃刀」の主張を情報理論を用いて数値化した物と見ることができる。

本研究で使用するモデルを MDL 基準を用いて選択する方法について検討する。アンケートの各質問に対応する X_i に関する学生 $j (j = 1, \dots, N)$ の回答を確率変数 X_{ij} で表すことにする。ここで、 N は学生の総数 (アンケートの有効回答数) を意味する。前節の考え方に基づき、 X_i 以外の質問内容 $X_{ck} (k = 1, \dots, l)$ に関する回答 X_{ck} が X_i の回答結果 X_{ij} へ及ぼす潜在的な影響を調べるために $P(X_{ij}|X_{c1j}, \dots, X_{clj})$ の確率分布に着目する。

アンケートに関する全ての回答結果 $\mathbf{D} = \{x_{ij}|i = 1, \dots, M, j = 1, \dots, N\}$ が学習データとして得られている下で、本モデルの MDL は以下で与えられる。

$$\begin{aligned}
 & -\log P(\mathbf{D}) + \frac{d \log N}{2} \\
 & = \sum_{i=1}^M \left\{ \frac{d_i \log N}{2} \right. \\
 & \quad \left. - \sum_{j=1}^N \log p_i(x_{ij}|x_{c1j}, x_{c2j}, \dots, x_{clj}) \right\} \quad (1)
 \end{aligned}$$

ここで $p_i(x_{ij}|x_{c1j}, \dots, x_{clj})$ は $P(X_{ij} = x_{ij}|X_{c1j} = x_{c1j}, \dots, X_{clj} = x_{clj})$ によって定義される確率分布関数であり、これらの値は条件付き確率表に対応する。また、 d_i は X_i 毎に定義される $P(X_i)$ 、あるいは $P(X_i|X_k) (i \neq k)$

に関する確率分布の自由度, そして $d = \sum_{i=1}^m d_i$ を示している。MDL では d は確率モデルのパラメータ数と説明されることが多いが, これは確率モデル自体の記述長に対応しており確率表を用いたベイジアンネットワークでは各ノードの自由度の合計に対応している。MDL が最も小さくなるように各ノード間の因果関係を選択することで, 学習データの下で最適なネットワーク構造が選択できる。

MDL は小さな数値を示すほど良い確率モデルであることを意味しているので, 評価関数として使用する場合には MDL の符号を逆にして使用する (Suzuki, 1993)。ちなみに MDL の符号を逆にしたものには BIC (Bayesian information criterion) という名前があり, 統計的推測の観点からまったく別の計算方法により導き出されたものと完全に一致する。

良いネットワークを判断するための評価関数が決定すれば, 残る問題はこの評価関数を最大にするネットワークの結合の仕方を見付け出すことになる。しかし, 多くのノードを含むベイジアンネットワークのネットワークの構造が取り得る場合の数は膨大になるために全数探索などの手法では計算機を用いたとしても現実的な時間で答を出すことは不可能である。そこで, 機械学習の手法を用いて評価関数を最大とするネットワーク構造の近似値を求めることになる。本研究では焼き鈍し法 (シミュレーテッドアニーリング法) を用いてこの近似値を求める。

焼き鈍し法は, 金属材料を熱した後で徐々に冷やし, 結晶を成長させてその欠陥を減らす作業により内部エネルギーを最小とする結晶構造を作り出す現象を模倣することで, 評価関数を最大にする組み合わせを見つけ出す手法である。この方法のポイントとなるのは, 高温の状態から温度を下げていく時のコントロールにある。本研究でもパラメータを変更しながら何回もシミュレーションを行うことで最終結果を導き出している。

焼き鈍し法によりベイジアンネットワークの構造を決定する方法は以下ようになる。

- (1) 初期温度 t を設定する
- (2) 適当な初期ネットワーク (本研究では結合がまったく無いネットワーク) を用意する
- (3) 評価値として現在のネットワークの MDL 値を計算しておく
- (4) 乱数を用いて 2 つのノード a と b を選び

出す。

- (5) a と b の間の結合を変更する
 - $a \rightarrow b$ の間にすでに結合が存在していれば削除する。もしくは $a \leftarrow b$ のように逆転する
 - $a \rightarrow b$ の結合が存在していなければ追加する。ただし, その追加した結合によりネットワーク中にループが生じる場合には変更をキャンセルし, (4) に戻る。(ベイジアンネットワークの確率的な要請による)
- (6) 新しい結合で MDL 値を計算する。
- (7) 「古い MDL 値」と「新しい MDL 値+乱数」を比べて後者の値が大きい場合は a と b の間の結合の変更をキャンセルし (4) に戻る。ここで「乱数」は温度 t の数値が高いほどランダム性が高くなるようにしておく。
- (8) 繰り返しの処理の中で結合の変更がキャンセルされることが長い間繰り返された場合はアルゴリズムを終了する。
- (9) 温度 t の値を少し減らす。
- (10) (4) に戻る

本ソフトウェアでは, このアルゴリズムを実行する時にネットワークの構造の変化をリアルタイムで表示させることも可能となっている。初期は温度 t が高いので, 非常に多くの結合が生成されて結合と切断が繰り返されるが, 温度 t が低くなるにしたがって結合数が少なくなっていき, 変化も少なくなってくる様子が見られる。

4. 結果

アンケートは Google フォームを用いて作成し, 北海道情報大学の学生に回答を呼び掛け, 2018 年 11 月 7 日から 2018 年 11 月 16 日までの 10 日間に回答を受け付けた。その結果, 合計で 115 件の回答を得ることができた。

ベイジアンネットワークを用いた分析結果の前に, プログラミングに対する自信と, その他の質問項目との統計的な関係についての結果を示す。

4-1 大学の科目, または独学でプログラミングを学んでいることとプログラミングに対する自信の関係

大学の科目, または独学でプログラミングを学んでいることとプログラミングに対する自

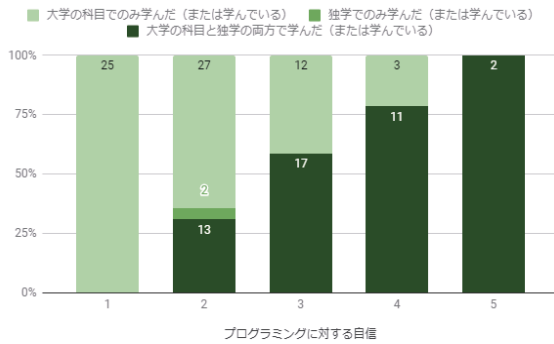


図3 大学の科目、または独学でプログラミングを学んでいることとプログラミングに対する自信の関係

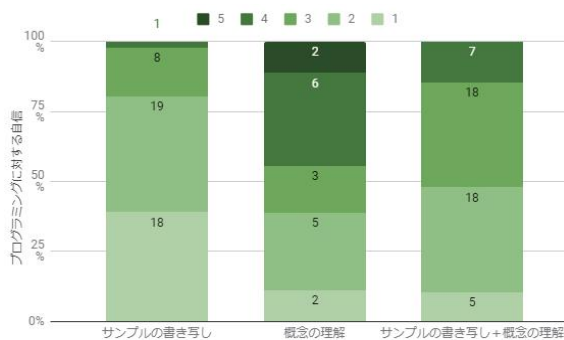


図4 プログラミング学習スタイルとプログラミングに対する自信

信の関係は図3のようになった。

「独学のみ」「大学の科目のみ」「大学の科目と独学の両方」という順序尺度として捉えたときに、大学の科目、または独学でプログラミングを学んでいることとプログラミングに対する自信の相関係数を求めたところ中程度の正の相関が認められた ($r = 0.53$)。

この結果から、独学のみ、または大学の科目のみでプログラミングを学んでいる者よりも、大学の科目と独学の両方でプログラミングを学んでいる者の方がプログラミングに対して自信を持っている傾向が見受けられることがわかる。

4-2 プログラミングの学習スタイルとプログラミングに対する自信の関係

プログラミングの学習スタイルとプログラミングに対する自信の関係は図4のようになった。

図4のグラフの系列の色の濃さが「プログラミングに対する自信」を5段階評価した結果に対応している。

サンプルを書き写して動作させる学習スタイルでプログラミングを学んでいる者とプログラミングに対する自信の相関係数を求めたところ

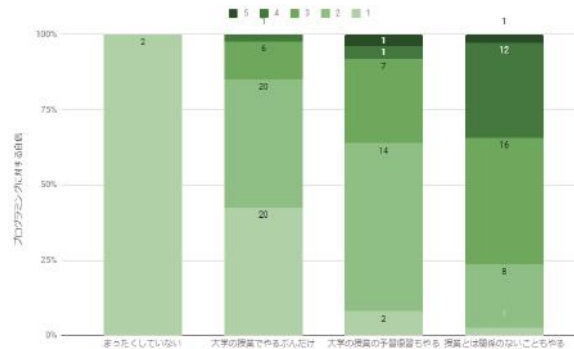


図5 プログラミングの学習時間とプログラミングに対する自信

中程度の負の相関が認められた ($r = -0.43$)。

プログラミング言語の仕様の概念を理解しながら学んでいる者とプログラミングに対する自信の相関係数を求めたところ弱い正の相関が認められた ($r = 0.28$)。

この結果から、サンプルを書き写して動作させる学習スタイルでプログラミングを学んでいる者よりも、プログラミング言語の仕様の概念の理解しながら学ぶスタイルで学んでいる者の方がプログラミングに対して自信を持っている傾向が多少は見受けられることがわかる。また、サンプルを書き写して動作させる学習スタイルである場合、プログラミングに対する自信に良い影響を与えない可能性も示唆される。

4-3 プログラミングの学習時間とプログラミングに対する自信の関係

プログラミングの学習時間とプログラミングに対する自信の関係は図5のようになった。

図4と同様、グラフの系列の色の濃さが「プログラミングに対する自信」を5段階評価した結果に対応している。

プログラミングの学習時間とプログラミングに対する自信の相関係数を求めたところ中程度の正の相関が認められた ($r = 0.63$)。

この結果から、プログラミングの学習時間が多いほど、プログラミングに対して自信を持つ傾向にあることがわかる。

4-4 プログラムの自作経験とプログラミングに対する自信の関係

大学の授業とは関係のないプログラムを自分で作った経験の有無とプログラミングに対する自信の関係は図6のようになった。

プログラムの自作の経験の有無とプログラミングに対する自信の相関係数を求めたところ

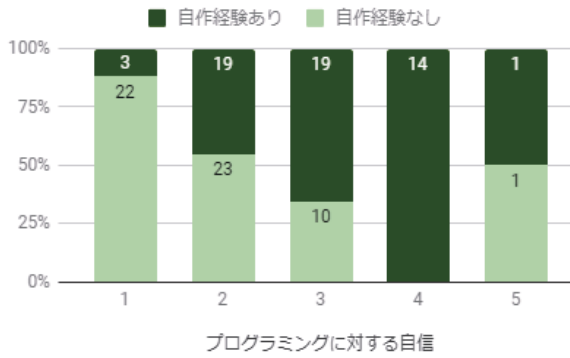


図6 プログラムの自作経験とプログラミングに対する自信の関係

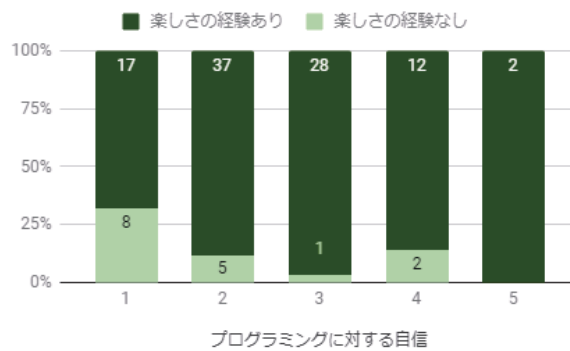


図7 プログラミングの授業に対する楽しさの経験の有無とプログラミングに対する自信の関係

る中程度の正の相関が認められた ($r = 0.49$)。この結果から、大学の授業とは関係のないプログラムを自分で作った経験がある者ほど、プログラミングに対して自信を持つ傾向が多少見受けられることがわかる。

4-5 プログラミングの授業に対しての楽しさの経験とプログラミングに対する自信の関係

プログラミングの授業に対して楽しさを感じた経験の有無とプログラミングに対する自信の関係は図7のようになった。

プログラミングの授業に対して楽しさを感じた経験の有無とプログラミングに対する自信の相関係数を求めたところ弱い正の相関が認められた ($r = 0.21$)。

この結果から、プログラミングの授業に対して楽しさを感じたことのある者の方がプログラミングに対して自信を持つ傾向が多少は見受けられることがわかるが、それほど強い関係があるような結果とはなっていない。

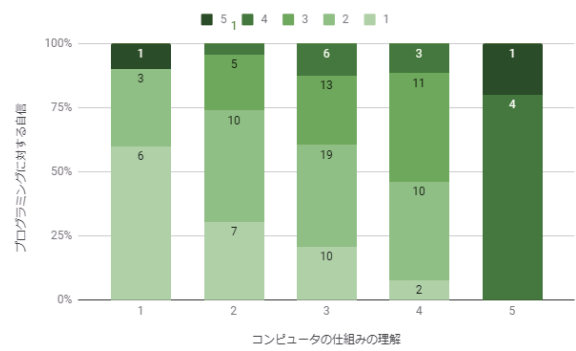


図8 コンピュータの仕組みの理解度とプログラミングに対する自信の関係

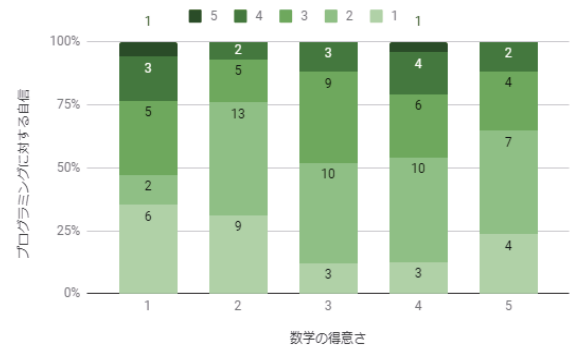


図9 数学の得意さとプログラミングに対する自信の関係

4-6 コンピュータの仕組みの理解度とプログラミングに対する自信の関係

コンピュータの仕組みの理解度とプログラミングに対する自信の関係は図8のようになった。

コンピュータの仕組みの理解度とプログラミングに対する自信の相関係数を求めたところ中程度の正の相関が認められた ($r = 0.41$)。

この結果から、コンピュータの仕組みをより理解しているほど、プログラミングに対して自信を持っている傾向が見受けられることがわかる。

4-7 数学の得意さとプログラミングに対する自信の関係

数学の得意さとプログラミングに対する自信の関係は図9のようになった。

数学の得意さとプログラミングに対する自信の相関係数を求めたところほとんど関係がないことが認められた ($r = 0.05$)。

4-8 英語の得意さとプログラミングに対する自信の関係

英語の得意さとプログラミングに対する自信の関係は図10のようになった。

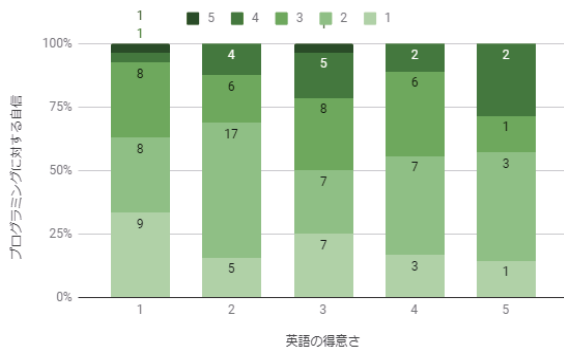


図 10 英語の得意さとプログラミングに対する自信の関係

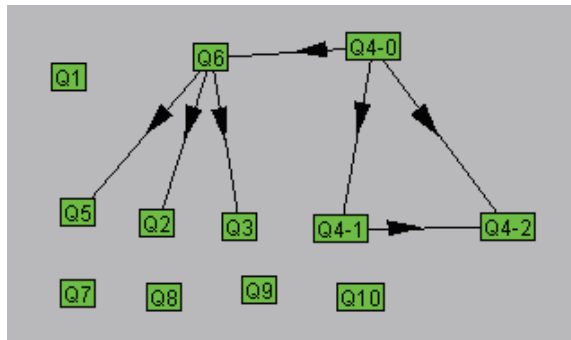


図 11 生成されたベイジアンネットワーク

英語の得意さとプログラミングに対する自信の相関係数を求めたところほとんど関係がないことが認められた ($r = 0.12$)。

4-9 ベイジアンネットワークを用いた分析結果

ベイジアンネットワークの構造を決定するために学生から集めたアンケートの情報を用い、焼き鈍し法のパラメーターを色々に変更しながら計5回の機械学習を行わせたが、5回中4回で等価なネットワークが出力された。ここで等価なネットワークとはMDLの値が同じで、ノード間の連結の方向だけが一部異なるネットワークのことである。ベイジアンネットワークのノード間の連結は向きのある矢印であり、矢印の始点が原因で矢印の終点が結果と見做すことができるが、そもそも確率的な従属性だけでは、原因と結果の区別ができない場合があるので、このような等価なネットワークが複数生成されるということが起る。そのような等価なネットワークの中の一つを図11に示す。

このベイジアンネットワークについて分析を行った。このネットワークでは以下の箇条書きに示したノード間に確率的な関係がある

ということになる。ベイジアンネットワークにおける因果関係は、相関係数により測られる統計的な相関関係とは異なる物である。そして本研究ではベイジアンネットワークとMDL原理により選択された結合が重要であるという立場ではあるが、その関係性を数値的に示すためには確率表などが必要である。ここでは相関係数も参考に示しているがあくまで参考として参照いただきたい。

- Q4-0 ⇔ Q4-1 サンプルを書き写して動作させる学習スタイルと、プログラミング言語の仕様の概念を理解していく学習スタイル ($r = -0.34$)
- Q4-0 ⇔ Q4-2 サンプルを書き写して動作させる学習スタイルと、独自の学習スタイル(自由記述) ($r = -0.74$)
- Q4-1 ⇔ Q4-2 プログラミング言語の仕様の概念を理解していく学習スタイルと、独自の学習スタイル(自由記述) ($r = -0.39$)
- Q4-0 ⇔ Q6 サンプルを書き写して動作させる学習スタイルと、大学の授業とは関係の無いプログラムを作ったことがあるということ ($r = -0.29$)
- Q6 ⇔ Q2 大学の授業とは関係の無いプログラムを作ったことがあるということと、プログラミングを大学科目・独学で学んでいること ($r = -0.53$)
- Q6 ⇔ Q3 大学の授業とは関係の無いプログラムを作ったことがあるということと、プログラミングに自信があること ($r = 0.49$)
- Q6 ⇔ Q5 大学の授業とは関係の無いプログラムを作ったことがあるということと、プログラミングの勉強に多くの時間を割くこと ($r = 0.66$)

ベイジアンネットワークのノード間に結合が生成されるかどうかはMDL原理によって決まる物で、統計学の相関係数とは異なる考えに基いているが、おおよそ相関係数の絶対値が大きい所に結合が生成されていることが確認できた。Q4は複数の選択を選ぶことができる質問であり、これらの中に強い相関があるのは質問の形式によるところが大きいと考えられる。上記7つのノード間以外には結合が生成されていないので、MDL原理の主張にもとづけばそれらの中には関係が無いと見做すべきであると言える。

統計的な相関関係は、因果関係とは別の物であり混同しないことが重要である。実際今回

のベイジアンネットワークにおいても例えば Q4-0 → Q6 の結合を Q4-0 ← Q6 というように逆向きにしても同時確率分布や MDL の値に違いは生じない。しかし中には矢印の方向を変えると確率分布として異なる物になり、MDL の値も変化するような矢印が存在する。これらの間には矢印の始点を原因、終点を結果とした因果関係が成立する可能性があることを示唆している。つまり、そのように考えた方が説明がつく、物事が単純に捉えられる、予測が上手くいく、という状況である。今回のネットワークの場合は Q6 から Q2, Q3, Q5 に接合されている矢印がこれに該当する。つまり、大学の授業とは関係のないプログラムを作ることが、プログラミングを大学科目・独学で学んでいることと、プログラミングに自信があること、プログラミングの勉強に多くの時間を割くことに影響を及ぼしているという可能性を示している。

5. 考察

5-1 プログラムの自作経験と、大学の科目、または独学でプログラミングを学ぶこととの関係

本研究の分析により、大学の授業とは関係のないプログラムを自分で作った経験が、大学の科目か、または独学でプログラミングを学ぶかという結果につながる原因であるという可能性が示唆された。

相関としては、プログラムの自作経験があるほど大学の科目に加えて独学でもプログラミングを学んでいるという関係になっている。自らプログラムを作るという行為が原因となり、その結果、独学でプログラミングを学ぶという関係と捉えれば素直に理解できる因果関係である。ベイジアンネットワークでは全体が有機的に連動しているため、一部の連結だけを入れ替えただけの MDL 値の比較は意味を持たない場合があるが、実際に該当する部分の矢印を反転させて MDL 値を計算してみると、もともと 0.984 だったものが反転させたことにより 0.991 となり 0.007 上昇する。数値としては小さいが優位な差が出ていると言える。

注目すべき点として、この逆の因果関係は今回の分析では見出されていないため、独学でプログラミングを学ぶという行為が自らプログラムを作るという行為につながる可能性を示唆しない点にある。

5-2 プログラムの自作経験とプログラミングの学習時間の関係

本研究の分析により、大学の授業とは関係のないプログラムを自分で作った経験が、プログラミングの学習時間という結果につながる原因であるという可能性が示唆された。

相関としては、プログラムの自作経験があるほど大学の授業時間以外でもプログラミングの学習に時間を割いている、という関係になっている。

アンケートでの質問の内容では、授業時間以外を使ってプログラミング授業の予習復習を行うか、という点についても評価しているため、プログラミングの授業に対する姿勢や熱心さ、という結果にも影響を与えている関係が見いだせる可能性がある。

5-3 プログラムの自作経験とプログラミングに対する自信の関係

本研究の分析により、大学の授業とは関係のないプログラムを自分で作った経験が、プログラミングに対する自信という結果につながる原因であるという可能性が示唆された。

大学の授業等でのプログラミングの作業は指示された内容に基づいて受動的な姿勢でプログラムを作ることとなり、そこに自主性や創造性が発揮されないことが考えられる。一方、大学の授業とは関係のないプログラムを自分で作るという行為は、自主的かつ能動的であり、自ら必要性を見出した上での行為である。この自主性を伴ったプログラミング経験が必然的にプログラミング学習の動機を作り、学習量やプログラミングスキルの獲得に貢献している可能性が考えられる。

しかし、自主性の強要は必然的にできないため(強要した時点で、それは自主的な行為とは言えなくなる)、教育の方法論として、自主性の発揮を直接指示するなどの方法は賢明ではない。

5-4 ベイジアンネットワークを用いた分析の有用性

本研究のベイジアンネットワークを用いた分析で得られた因果関係の分析結果は、前述の考察でも述べたように、意味的に理解可能な結果となった。

また、ベイジアンネットワークを用いることにより、相関はしているが因果関係として認

められない関係についても推論することができた。

6. 課題

6-1 自主的なプログラミング活動に影響を与える原因

考察で述べたように、自主的にプログラムを作る経験によりプログラミングに対する自信や、プログラミングの学習時間や学習態度に好影響を与える可能性を考えた。

しかし、どのような原因が自主性を伴ったプログラミング行為につながるのか、という因果関係については分析の余地を残している。

6-2 本研究で取り上げなかった要因

本研究で使用したアンケートの項目以外にも、プログラミングに対する自信に影響を与える要因の存在の可能性は十分考えられる。

6-3 プログラミングに対する自信と実力の差

本研究で実施したアンケートでは、回答者の主観的なプログラミングに対する自信を回答してもらったが、実際のプログラミングの実力は測っていない。本研究で実施した自己申告による自信の程度を使用した場合の分析と、何らかの方法で実力の計測をした結果を使用した場合では、因果関係の分析結果に違いが出る可能性を残している。

参考文献

- [1] 韓太舜・小林欣吾 (1994) 『情報と符号化の数理』, 岩波書店, 東京千代田区。
- [2] Heckerman, David (1995) "A Tutorial on Learning With Bayesian network," March, Technical Report:Microsoft Research Advanced Technology Division.
- [3] 本村陽一 (2003) 「ベイジアンネットワークによる確率的推論技術」, 『計測と制御』, 第42巻, 第8号, pp.649-654。
- [4] 岡本雅子・喜多一 (2014) 「プログラミングの「写経型学習」における初学者のつまずきの類型化とその考察, パイディア, 滋賀大学教育実践研究指導センター紀要」, 『パイディア: 滋賀大学教育学部附属教育実践総合センター紀要: memoirs of the Center for Educational Research and Training, Shiga University』, 第22巻, pp.49-53。
- [5] 岡本雅子・村上正行・吉川直人・喜多一 (2013) 「<実践報告>プログラミングの写経型学習過程を対象としたつまずきの分析とテキスト教材の改善: 作業の自立的遂行と作業を介した理解のための支援と工夫」。
- [6] Pearl, Judea (1988) *PROBABLISTIC REASONING IN INTELLIGENT SYSTEMS: Networks of Plausible Inference*, San Francisco. California: Morgan Kaufmann Publishers, Inc.
- [7] Russell, S. and P. Norvig (1997) 『エージェントアプローチ人工知能』, 共立出版, (古川康一訳)。
- [8] Suzuki, Joe (1993) "A construction of Bayesian networks from database on an MDL principle," *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 266-273.