

正例および負例の集合を考慮した科目分類支援システムの提案と
経験強化型学習との融合

Proposal of the Active Course Classification Support System with Positive and
Negative Examples and Combining with Exploitation-oriented Learning

宮崎 和光, 井田 正明
MIYAZAKI Kazuteru, IDA Masaaki

1. はじめに	3
2. NIAD-UE の科目審査の概要と科目分類支援システム	4
2.1 現状の科目審査プロセスの概要	4
2.2 科目分類支援システムの概要	6
3. 経験強化型学習を組み込んだ ACCS	7
3.1 基本的な考え方	7
3.2 ACCS における経験強化型学習の活用	8
3.3 経験強化型学習を組み込んだ ACCS の全体像	9
4. 正例および負例の集合を考慮した ACCS の提案	9
4.1 ACCS with XoL の問題点	9
4.2 正例および負例集合の作成	10
4.3 経験強化型学習利用下での正例および負例の集合を考慮した ACCS	10
5. 評価実験	11
5.1 学習方法	11
5.2 学習結果	12
5.3 考察	12
6. おわりに	13
謝辞	13
参考文献	13
ABSTRACT	15

正例および負例の集合を考慮した科目分類支援システムの提案と 経験強化型学習との融合

宮崎 和光*, 井田 正明*

要 旨

大学評価・学位授与機構 (NIAD-UE) では, 短期大学・高等専門学校卒業生及び専門学校修了者等を対象に, 単位累積加算を基にした学士の学位授与事業を行っている。この制度の利用を希望する者は, 自らの判断で自らが修得した科目を NIAD-UE が定める科目区分に合致するように分類・整理し申告する必要がある。それに対し, NIAD-UE では, 申請者による分類の正しさを, 学位審査会に所属する各専門分野の専門委員が, NIAD-UE によって公式に設定された各科目区分ごとの「専門科目名のリスト」を手がかりに, 申告された科目のシラバスを読むことで検討している。これまでに著者らは, 専門委員の分類作業を支援するシステムとして, 科目分類支援システム (Course Classification Support System; CCS) [11] および分類候補数の能動的調整を可能にした科目分類支援システム (Active Course Classification Support System; ACCS) を提案してきた [12]。さらに, 近年は, ACCS に機械学習手法の 1 種である「経験強化型学習」 (Exploitation-oriented Learning; XoL) [15] を組み込んだ手法である ACCS with XoL を提案し, 情報工学区分を対象に, 提案手法の有効性を確認している [16, 17]。しかしそこでの ACCS の利用方法は, XoL の特性を考慮したものにはなっていない。そこで, 本論文では, 過去の判定事例から正例および負例集合を作成し, その集合に基づく ACCS の利用方法を提案する。さらに, 文献 [16, 17] で提案された ACCS with XoL と組み合わせ, 情報工学区分を対象に, 提案手法の有効性を確認する。

キーワード

シラバス, 科目分類, 学位授与事業, 推薦システム, 経験強化型学習, 強化学習

1. はじめに

大学評価・学位授与機構 (NIAD-UE) では, 短期大学・高等専門学校卒業生及び専門学校修了者等を対象に, 単位累積加算を基にした学士の学位授与事業を行っている。この制度を利用し学士の学位授与を希望する者は, 各専門分野 (専攻の区分という) ごとに定められた所定の単位数を修得し, 修得単位に関する審査の基準を満たさなければならない。申請者は, 自らの判断で修得した科目を NIAD-UE が定める科目区分に合致するように分類・整理し申告する。それに対し, NIAD-UE では, 申請者による分類の正しさを, 各専門分野の専門委員が, 各科目区分ごとに設定されている

「専門科目名のリスト」を手がかりに, 申告された科目のシラバスを読むことで検討している。

シラバスは学生に対する授業の選択と履修時のガイドのための情報提供を本来の目的とするが [7], 授業内容を最も詳しく示す資料として大学評価の支援や大学における教育情報の活用などにも積極的に活用されてきている [4]。高等教育の質保証の観点から効果的なシラバスの作成方法や活用方法についての研究や, シラバスを電子テキストの分析対象とみなして情報工学的アプローチからの研究も増加してきている [21, 3, 22, 5]。

情報工学的な研究としては, シラバスのインターネットからの自動収集ならびにそこからの情報抽出 [6, 26, 31] や, シラバスの文書解析による

* 独立行政法人 大学評価・学位授与機構 研究開発部

教育課程の比較研究がなされてきた [21, 3, 22, 5]。教育課程の比較分析の手順はつぎのようになる：シラバスの収集とデータベース化，分析対象とする教育課程とシラバスの選択，シラバス文書の単語情報の分析（形態素解析），専門用語の抽出，専門用語に基づき各シラバスの内容の定量化，シラバス間の類似度の計算，類似度に基づきクラスタリングや多変量解析の実施，結果の可視化，近接性の把握による教育課程の比較分析（結果解釈）。具体的な分析として，工学系，情報系などの分野についてシラバスによる教育課程の比較分析が行われてきた。近年においてもこのようなシラバスの文書解析は，各大学等のシラバス集合を対象に実施されている（例えば，文献 [20, 27, 18] など）。

シラバスにオントロジーを用いた知識の記述による講義構造の構造化を行うことによって，より高度な検索を行える講義データベースの検討が行われてきた [29]。また，東京大学工学部では工学教育の講義等の科目を工学知の粒度と考え，シラバスを基にして工学系カリキュラムの構造化と可視化が行われてきた [23]。これにより知識（カリキュラム）の全体像を把握し，学生が進むべき道を自ら検討したり，知識の重複や不十分な個所を発見することなどが可能となる。現在，シラバス構造化システムが稼働している [25]。

本研究の目的は，学位授与事業の支援のために，シラバスをその内容に基づき分類することにある。NIAD-UE では，申請された科目のシラバスを，短期間に，数種類の科目区分に分類することが要求される。しかしながら，近年の申請者数の増大から，この作業には膨大な時間と労力を要しており，情報技術を活用した科目の分類支援が望まれている。このような NIAD-UE におけるシラバスの内容に基づく科目分類は，学位授与事業の持つ審査業務という性質から，申請者の不利益となるような判定を許容できないという性質を持つ。そのため情報技術による支援を行う際には，科目分類の主体はあくまで専門委員会にあり，システムには従の役割，すなわち**推薦システム** [28] としての振る舞いが期待されている。

このような推薦システムとしての観点から，著者らはこれまでに，各科目区分ごとに，その科目区分のみに含まれている特徴的な用語をキーワードとして抽出することで，専門委員の分類作業を

支援するシステムとして，「**科目分類支援システム (CCS)**」を提案してきた [11]。さらに，その発展として，シラバス内に存在する用語の各科目区分への結び付きの強さを連続値で評価することで，分類を試みている科目区分に帰属する可能性の高い順に申告された科目を専門委員に提示することができる「**分類候補数の能動的調整を可能にした科目分類支援システム (ACCS)**」を提案している [12]。ACCS を用いることにより，専門委員等が自ら能動的に判定すべき科目の候補数を調整することが可能になり，分類作業の大幅な負担軽減が期待できる。このように，我々が提案する科目分類支援システムは，シラバスに基づくテキスト情報処理，自然言語処理の実践的な応用例となっている。

さらに，近年は，ACCS に機械学習手法のひとつである「**経験強化型学習**」(Exploitation-oriented Learning; XoL) [15] を組み込んだ手法である **ACCS with XoL** を提案し，情報工学区分を対象に，提案手法の有効性を確認している [16, 17]。しかしそこでの ACCS の利用方法は，XoL の特性を考慮したものにはなっていない。そこで，本論文では，過去の判定事例から正例および負例集合を作成し，その集合に基づく ACCS の利用方法を提案する。さらに，文献 [16, 17] で提案された ACCS with XoL と組み合わせ，情報工学区分を対象に，提案手法の有効性を確認する。

2. NIAD-UE の科目審査の概要と科目分類支援システム

2.1 現状の科目審査プロセスの概要

NIAD-UE における学位授与事業の制度・申請方法などの詳細は，「新しい学士への途」 [9] に詳しく示されている。ここでは，その中から，科目分類に直接関係する部分である単位修得に着目した申請方法の概略を述べる。

単位修得の観点で，申請者が第一に行うべきことは，専攻の区分の選択である。平成25年度現在，NIAD-UE では，表 1 に示すような国語国文学から体育学まで全部で59個の専攻の区分を用意している。申請者はこれら59個の中から自らの学修に最も近いと思われる一区分を選択し，その区分の学士の学位を得るために必要な単位を NIAD-UE に申告する必要がある。

それに対し、NIAD-UEでは、申請者が修得した単位を専攻に係る授業科目の単位である「専攻に係る単位」と、それ以外の「専攻に係る単位以外の単位」のふたつに大きく分け審査を行う。特に「専攻に係る単位」は、「専門科目の単位」と呼ばれる専門的な内容の授業科目の単位と、「関連科目の単位」と呼ばれる専門に関連する授業科目の単位に区分され、それぞれについて修得すべき単位数などの単位修得の要件（「**修得単位の審査の基準**」という）が定められている。これらの専攻に係る授業科目は、授業科目の内容や授業の方法（講義、演習、実験など）により、「専攻に係る授業科目の区分」（以下では、簡単のため「**科目区**

分」と呼ぶ）として細分化されている。また、専攻の区分によっては、複数の「専攻に係る授業科目の区分」がまとまって「群」として示されている場合もある。

専攻の区分ごとの修得単位の審査の基準を示す表には、「専門科目」、「関連科目」、「群」、「科目区分」ごとに修得すべき単位数が示されている。また、各専攻の区分にとって特徴的かつ重要な「専門科目」の科目区分に対しては、分類作業を手助けするための情報として、「専門科目名のリスト」が設定されている。例として、表2に、平成25年度時点での、情報工学区分における「修得単位の審査の基準」を示す。

表1 専攻の区分

国語国文学	英語・英米文学	独語・独文学
仏語・仏文学	中国語・中国文学	ロシア語・ロシア文学
歴史学	哲学	心理学
宗教学	教育学	神学
社会学	社会福祉学	比較文化
地域研究	国際関係	科学技術研究
社会科学	法学	政治学
経済学	商学	経営学
数学・情報系	物理学・地学系	化学系
生物学系	総合理学	薬科学
看護学	検査技術科学	臨床工学
放射線技術科学	理学療法学	作業療法学
言語聴覚障害学	視能矯正学	鍼灸学
口腔保健衛生学	口腔保健技工学	栄養学
機械工学	電気電子工学	情報工学
応用化学	生物工学	材料工学
土木工学	建築学	社会システム工学
芸術工学	商船学	農学
水産学	家政学	音楽
美術	体育学	

表2 情報工学区分の「修得単位の審査の基準」（平成25年度版「新しい学士への途」より抜粋）

専門科目 (40単位以上)	A群 (30単位以上)	情報工学基礎理論に関する科目（4単位以上） 計算機システムに関する科目（4単位以上） 情報処理に関する科目（4単位以上） 情報に関連する科目	科目区分1 科目区分2 科目区分3 科目区分4
	B群	情報工学に関する演習・実験・実習科目	科目区分5
関連科目 (4単位以上)	工学の基礎となる科目 工学及び周辺技術等に関する科目		科目区分6
「専門科目」及び「関連科目」以外の科目			科目区分7

申請者は、修得した科目を、その内容により、各専攻の区分に係る「専門科目」、「関連科目」およびそれら以外の科目に自ら分類・整理しなければならない。これは、上述の情報工学区分の場合、表2の1～7のいずれかの科目区分に修得した科目を申請者自らが分類することを意味する。このように申請が自己申告を基本としているため、場合によっては、不適切な科目分類が行われる可能性があり、NIAD-UEにおいて、その適切性の確認が行われている。特に、専門科目の判定は、各分野に精通した専門家による判定が必要である。NIAD-UEでは、そのような専門性の高い科目分類の確認を支援するために、平成12年度から、過去の科目分類の判定結果をデータベース化し、「専攻の区分ごとの修得単位の審査の基準」の判定に活用している。このデータベースは科目データベース（科目DB）と呼ばれている。

科目DBに登録されている科目に関しては、通常、専門的な判断は必要とされない。それに対し、過去に判定例が存在しない科目（初出科目）については、専門委員会において、申請者による科目分類の正しさを、その科目のシラバスを読むことで検討している。初出科目に対する判定作業は、基本的には対応するシラバスをすべて読むことで行われており、近年の申請者数の増大から、その作業には膨大な労力と時間を要している。特に、専門委員会では、「専門科目名のリスト」が存在する科目区分に対する分類作業に多くの時間が費やされる。

そこで、著者らは、そのような分類作業に要する専門委員の作業量軽減を指向した手法としてCCS [11] や ACCS [12] といった科目分類支援システムを提案している。

2.2 科目分類支援システムの概要

2.2.1 CCS の概要

CCS および ACCS の要は、科目DBにある。文献 [11] では、この科目DBを加工することで、各科目区分ごとに、その区分のみに内包されている特徴的な用語の集合である $myDB = \{myDB(1), \dots, myDB(N)\}$ (N は科目区分の種類) を作成した。各 $myDB(i)$ ($i = 1, 2, \dots, N$) は、科目区分 i に分類されているシラバス内に存在する用語集合から科目区分 i と無関係な科目区分に分類されている用語

集合を取り除くことで作成される。ここで、各用語は文献 [19] の「接続頻度に基づく手法」により抽出する。なお、現在、この抽出作業は、文献 [19] に基づく Web サービスである「言選 Web」 [8] により行うことができる。

「科目分類支援システム (CCS)」では、科目区分 i に帰属する科目を探しているとき、申請者が申告している科目のシラバスと $myDB(i)$ とのマッチングをとる。その結果、各シラバスごとに、 $myDB(i)$ とマッチングした用語の集合が得られる。この集合を用いることで、 $myDB(i)$ とのマッチングの有無により、専門委員に提示する科目集合を限定することができる。そのため、従来の全科目を判定する場合に比べ、専門委員の作業量の軽減が期待できる。

ところで、CCS の $myDB$ は、各科目区分ごとに、その科目区分のみに含まれている用語を集めたものである。しかしそこでは、その用語と、その用語が含まれている科目数との関係は考慮されていない。本来、ある特定の用語が、ある科目区分にとって特徴的な語であるか否かは、その用語が、どれだけ多く、その科目区分に関係するシラバス内に出現したかに依存し変化するはずである。ある特定の用語とある科目区分の関係など、一般に概念間の関係性の強度は計量言語学の分野において様々な指標が検討されてきている。著者らはこれまでに、各用語とその用語が含まれている科目数との関係に注目し、「分類候補数の能動的調整を可能にした科目分類支援システム (ACCS)」を開発した。

2.2.2 ACCS の概要

ACCS でも、CCS 同様、まずはじめに、文献 [19] の手法によりシラバスから用語を抽出する。その後、その抽出された各用語が、各科目区分 i に関係するシラバス内に含まれる確率を区間推定する。ある用語が科目区分 i に分類可能なシラバス内に含まれる確率の信頼区間は、「その用語が含まれているシラバスの数」 (n) に対する「その用語が含まれており、かつ、科目区分が i であるシラバス数」 (x) の割合として以下の式から求められる [2]。

$$\frac{\frac{x}{n} + \frac{Z^2}{2n} \pm \frac{Z}{\sqrt{n}} \sqrt{\frac{x}{n} \left(1 - \frac{x}{n}\right) + \frac{Z^2}{4n}}}{1 + \frac{Z^2}{n}}, \quad (1)$$

ここで Z_γ は、信頼度95%のとき1.96、信頼度99%のとき2.58となる正規分布の両側パーセント点である。なお、5章の評価実験では、 $Z_\gamma = 1.96$ (信頼度95%) とした。

この計算結果、すなわち、「ある用語が、科目区分 i に関係する科目のシラバス内に存在する確率の区間推定値」を各科目区分ごとに**関係値**としてとりまとめ、[用語, 科目区分1に対する関係値, ..., 科目区分 N に対する関係値]の組を1エン트리とする **myDB_c** と呼ばれるDBを構築する。

myDBは、各科目区分のみに含まれている用語の集合により構成されるが、それは(1)式において、 $x = n$ 、すなわち、(1)式の上側信頼限界が1となる用語を集めたものと言える。そのような用語は、該当する科目区分との関係が強いことが期待できる一方、一般に、 n が大きくなるに連れ総数は減少する。そこで、myDB_cでは、(1)式を用いて、各用語の各科目区分との関係の強さを、より関係が強い用語ほどその上側信頼限界が1に近づくという形で評価することで、myDBにおける用語の総数の不足に対応している。

科目区分 i に帰属する科目を探しているときのACCSは、次のように構成される。

ACCS (科目区分 i に関する科目分類支援用)

step 0: 申請者が申告している科目のシラバスとmyDB_cとのマッチングをとる。その結果、各シラバスごとに、myDB_cとマッチングした用語と、その語の各科目区分に対する関係値が得られる。

step 1: 閾値 t ($0 \leq t < 1$) の初期値を設定する。

step 2: step 0で得られる関係値を参照し、「科目区分 i に分類可能なシラバス内に存在する確率の信頼区間」の下側信頼限界が t 以上となっている用語を含む科目のシラバスのうち未判定なものを、その用語とともに専門委員に提示し、判定作業を行っていただく。

step 3: 科目区分 i に課せられている単位数が満足されれば、その科目区分の判定を終える (必要に応じ、他の科目区分または群の判定に移る)。満足されない場合は、閾値を下げ step 2へ戻る。

myDB_cは、CCSのmyDBとは異なり、各用語と、その用語が含まれている科目数との関係を考慮に入れたDBとなる。また関係値を参照するこ

とで、各用語の各科目への結び付きの強さを連続的に評価することもできる。その結果、ACCSを用いることにより、専門委員等が自ら能動的に判定すべき科目の候補数を調整することが可能になり、分類作業の大幅な負担軽減が期待できる。

2.2.3 CCS および ACCS の問題点

その一方で、CCS, ACCSともに、いずれの科目区分から審査を行うべきかは明らかではない。現状の審査においては、科目区分は先験的に決められており、通常は、先頭の科目区分から順に審査される。それに対し、理想的には、科目分類支援システムが、専攻の区分ごとに異なることが予想される科目区分の特性を学習し、その特性を考慮した形で審査すべき科目区分を決定することが望ましい。

そのような学習は、強化学習 (RL) [30] や経験強化型学習 (XoL) [15] で実現可能であり、近年、ACCSとXoLを組み合わせた手法として次章で述べるACCS with XoLを提案している [16, 17]。

3. 経験強化型学習を組み込んだ ACCS

3.1 基本的な考え方

ACCSを用いれば、未知の科目に対し、その科目のシラバスに含まれる各用語の、各科目区分との関係の強さを**関係値**として知ることができる。特に、2.2.2節で示したACCSのアルゴリズムは、関係値の中で閾値 t 以上の値を有している科目区分を、その科目が分類される可能性の高い科目区分であるとみなす手法である。そのため、閾値 t を1.0から徐々に下げていくことで、審査の対象としている科目区分に分類される可能性の高い科目を順番に知ることができる。以下では、便宜上、この学習を「**閾値の学習**」と呼ぶ。

この「**閾値の学習**」が十分に適切に行われている場合、すなわち、「科目区分Aに分類される可能性の高い科目を知りたい」というときに、「その時点で科目区分Aに分類されるとシステムが主張する科目群の中で最大の関係値を有する科目、すなわち、2.2.2節 step 2で最初に (最大の閾値で) 発見される科目が、実際に科目区分Aに分類される科目である」ならば、いずれの科目区分から審査を開始してもよい。なぜならば、単位不足となっている科目区分を次々充足させていけば、必要最

低限の科目審査数でその専攻の区分における修得単位の審査の基準を満たすことができるためである。

しかし実際には、「閾値の学習」は必ずしも適切なものとはならない可能性がある。すなわち、ある時点で科目区分Aに分類されるとシステムが主張する科目群の中で最大の閾値で発見された科目が、必ずしも科目区分Aに分類されるとは限らない。例えば、情報工学区分では、「計算機システム」(科目区分2)に分類される科目は比較的多く存在するが、「情報処理」(科目区分3)に関する科目は数が少ない傾向にある。そのため、ACCSにより、科目区分3に分類される可能性が高いとされた科目の中には、実際には、科目区分2などに分類される場合があり、本来の「情報処理」科目を発見する妨げになることがある。

このような「閾値の学習」の不完全さは、一般には、専攻の区分、さらには、その中の科目区分の設計に依存する。NIAD-UEでは、適宜、専門委員会において、この科目区分設計の見直しを行っているが、変更する際には、数年の移行期間を設けるなどの十分な配慮を行う必要がある。そこで通常は、各専攻の区分ごとに、科目区分設計の特性を学習し分類支援に活用することで、効率よく修得単位の審査の基準を満たすことが特に重要となる。例えば、情報工学区分では、「情報基礎」(科目区分1)や「計算機システム」(科目区分2)よりも優先して「情報処理」(科目区分3)に分類される科目を重点的に探すことが可能になることが期待される。次節では、そのための具体的方法について述べる。

3.2 ACCSにおける経験強化型学習の活用

各専攻の区分ごとに、科目区分設計の特性を経験強化型学習で学習し、専門委員に審査を依頼する際の科目区分の決定に活かすことを考える。これはすなわち、2.2.2節で述べた「ACCS (科目区分*i*に関する科目分類支援用)」における科目区分番号*i*の決定方法を与える学習システムを作成することを意味する。

科目区分設計の特性を学習するのに適した機械学習手法として、強化学習および経験強化型学習がある。ここでは、一般に、各**状態入力**に対し、出力すべき**行動**を**報酬**という特別な入力を手がかりに学習する。ここで、ある状態入力で出力すべ

き行動を与える関数を**政策**と呼ぶ。政策は、一般には、if-then型のルール形式「if 状態入力 then 行動出力」で記述される。例えば、今、3種類の状態入力 {状態A, 状態B, 状態C}, 4種類の行動出力 {行動1, 行動2, 行動3, 行動4} が存在した場合に、「状態Aでは行動3, 状態Bでは行動1, 状態Cでは行動2」を選択すべきと判断されたとき、政策は、{「if 状態A then 行動3」, 「if 状態B then 行動1」, 「if 状態C then 行動2」}といった3個のルールの集合として記述される。強化学習では、主として、各ルールに**重み**と呼ばれる評価値を付加し、その評価値を更新することで政策の学習が行われる。一方、XoLでは、重みを利用する手法も存在する[13]が、基本的には、ルールを直接分類することによって学習が進行する。

以下では、ACCSとの融合に際し必要となるこれら「状態入力」「行動出力」「報酬」の設定方法を順次述べる。

3.2.1 状態入力および行動出力

ある時点で審査の対象とすべき科目区分は、その時点での、各科目区分に課せられている必要単位数の満たされ具合に応じて変化させるべきである。そこで、文献[16, 17]では、そのような各科目区分の必要単位数の満たされ具合、すなわち、「修得単位の審査の基準を満たすために必要な単位数」を状態入力とする方法が採用されている。

一方、行動出力は、各状態入力ごとに、どの科目区分の充足を目指すか、すなわち、「修得単位の審査の基準を満たしたい科目区分番号」である。

3.2.2 報酬設計

強化学習および経験強化型学習では、報酬という結果の善し悪しのみを与える教師信号により学習を行う。報酬の設計は、これらの学習の成否に大きな影響を与える。

本問題設定において最も基本的な報酬設計は、「修得単位の審査の基準が満たされた時点で報酬を与える」というものである。しかし、そのような設計では、専門委員に依頼する科目数を減らす効果は期待できない。そこで、文献[16, 17]では、「各申請者ごとに最少の科目提示数で修得単位の審査の基準を満たした時点で報酬を与える」方法が採用されている。

3.3 経験強化型学習を組み込んだ ACCS の全体像

ACCS に前節で述べた経験強化型学習を組み込んだ手法は、ACCS with XoL と呼ばれている。ACCS with XoL の全体像を図 1 に示す。ここで、学習器とは強化学習または経験強化型学習に相当する。学習器への状態入力、各科目区分の基準を満たすために必要な単位数である。学習器は、その状態入力下で選択すべき行動（すなわち、修得単位の審査の基準を満たすべき科目区分番号）を自身の政策に従い決定する。学習器の出力は、そのまま ACCS の入力となる。ACCS は、入力された科目区分番号において、その中で最も高い閾値で発見された科目、すなわち最大の関係値を有している科目を専門委員に提示する。専門委員は、入力された科目が、いずれの科目区分に分類されるかを審査判定する。その結果に基づき各科目区分の必要単位数が計算され、3.2.1 節で述べたように、学習器への状態入力となる。また、同時に学習器へは、3.2.2 節で述べたような形で報酬が計算され与えられる。

学習データとしては、各科目の分類が既知である申請者のデータを用いる。そのデータを用いて、各状態入力下で選択すべき行動出力を報酬をもとに学習を行う。学習結果は、先に述べた if-then 型のルール形式の政策として記憶される。なお、一般に、強化学習や経験強化型学習では、多様な行動を出力することでより適切な政策の学習を目指す。そのためにここでも、学習時には、一様ランダムな確率で行動を選択し続けるランダム選択の採用が想定されている。

政策は、一度は、申請者ごとに保存される。複数の申請者分の学習データが存在する場合には、

各申請者ごとに、政策を各状態入力ごとに正規化し、複数人の政策が得られた時点で、それらを統合する。その際、各申請者の学習を等しく反映させるために、予め申請者ごとに政策は正規化しておく。最後に、統合された政策を用いて、未知の申請者に対応する。

なお、図 1 から明らかなように学習器の出力は、ACCS の性能に大きく依存する。一般に、ACCS の出力は、「修得単位の審査の基準を満たすために必要な単位数」以外の要素、例えば、閾値の値によっても変化する。したがって、学習器の観点では、本問題は、環境（学習器以外の部分として定義される）に、**不完全知覚** (perceptual aliasing) [1] が存在する問題として認識される。そのため不完全知覚問題が存在しない、例えば、**マルコフ決定過程** (Markov Decision Process; **MDP**) を対象とする各種の**動的計画法** (Dynamic Programming; **DP**) に基づく手法 [30] では、本問題は、解決困難な問題となっている点に注意されたい。

4. 正例および負例の集合を考慮した ACCS の提案

4.1 ACCS with XoL の問題点

ACCS with XoL は、ACCS に学習機能を付加したという意味で画期的な手法ではあるが、そこでの ACCS の利用形態は、従来単独で ACCS を利用していたときとまったく同一であり、XoL の特性を考慮したものになってはいない。すなわち、XoL が指定した科目区分に関する最大の閾値を有する科目を専門委員に提示する際に、他の科目区分の状態を一切考慮していない点が問題であると言える。

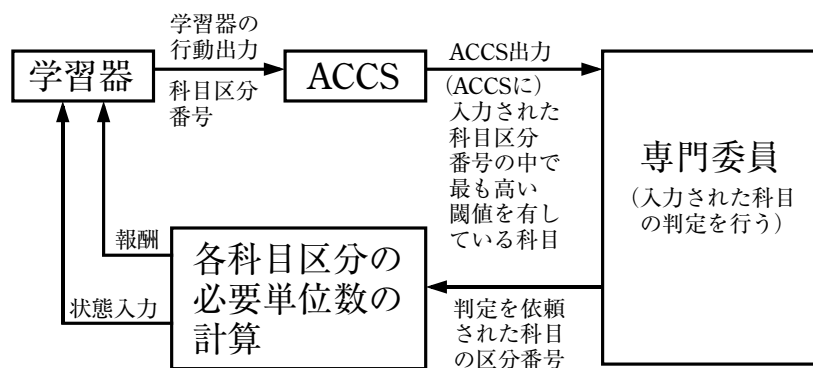


図 1 ACCS with XoL の全体像

そこで、本論文では、過去の判定事例から、注目する科目区分に対する正例および負例の集合を作成することで科目区分全体を考慮した ACCS の利用を実現する。

4.2 正例および負例集合の作成

4.2.1 閾値パターン

正例および負例集合を活用した ACCS with XoL を構成するために、まず初めに、各科目ごとに、その科目の各科目区分に関するマッチ具合を表す閾値を閾値パターンとして求める。この情報は、ACCS に各科目を入力することで得られる。

例えば、科目 A の科目区分 1 に関する閾値が 0.8、科目区分 2 に関する閾値が 0.5、科目区分 3 に関する閾値が 0.2、科目区分 4 に関する閾値が 0.3 であると ACCS から得られた場合、科目 A の閾値パターンは (0.8, 0.5, 0.2, 0.3) となる。

閾値は、2.2.2 節に示した ACCS のアルゴリズムの step 3 において、閾値を下げる際の刻み幅を固定することで、離散的な値をとることができる。それに対し、一般には、連続的な値をとる閾値を用いて閾値パターンを構成することも考えられる。しかしその場合には、状態空間が膨大になり、後の学習が困難となる恐れがあるため、ここでは閾値を利用する方法を採用した。

4.2.2 正例および負例集合

過去に専門委員会で判定されたことのある科目の閾値パターンを用いて、以下の手順で、各科目区分ごとに、**正例の集合**および**負例の集合**を作成する。作成方針には、種々のものが考えられるが、ここでは、過去の判定事例から、確実に言えると考えられる正例および負例の集合のみを作成する方針を採用した。

既に専門委員による分類結果が判明している科目に対し、正解、すなわち専門委員が分類した科目区分に対しては、その科目区分の正例の集合に、その科目の閾値パターンおよび、その閾値パターン中の、正解の科目区分の閾値をその閾値パターンの値から閾値 1.0 まで増加させたもの、および正解でない科目区分の閾値をその閾値パターンの値から閾値 0.0 まで減少させたものすべての組み合わせを登録する。なお、この登録に際しては、刻み幅（5 章の評価実験では 0.1）を設定し、離散的

に登録する。

一方、正解と異なる科目区分に対しては、その科目区分の負例の集合に、その科目の閾値パターンおよび、その閾値パターン中の、その科目区分の閾値をその閾値パターンの値から閾値 0.0 まで減少させたもの、および、それ以外の科目区分の閾値をその閾値パターンの値から閾値 1.0 まで増加させたものすべての組み合わせを登録する。

例えば、閾値パターン (0.8, 0.5, 0.2, 0.3) の科目が、科目区分 1 に分類されていたとすると、科目区分 1 の正例の集合に、まず、{(0.8, 0.5, 0.2, 0.3), (0.9, 0.5, 0.2, 0.3), (1.0, 0.5, 0.2, 0.3)} を登録し、この各々に対し、科目区分 2 を 0.5 から 0.0 まで減少させたもの、科目区分 3 を 0.2 から 0.0 まで減少させたもの、科目区分 4 を 0.3 から 0.0 まで減少させたものすべての組み合わせを適用したものを登録する。すなわち、{(0.8, 0.4, 0.2, 0.3), ..., (0.8, 0.0, 0.2, 0.3)} の 5 パターン、{(0.8, 0.5, 0.1, 0.3), (0.8, 0.5, 0.0, 0.3)} の 2 パターン、{(0.8, 0.5, 0.2, 0.2), ..., (0.8, 0.5, 0.2, 0.0)} の 3 パターン、および {(0.9, 0.4, 0.2, 0.3), ..., (0.9, 0.0, 0.2, 0.3)} の 5 パターン、{(0.9, 0.5, 0.1, 0.3), (0.9, 0.5, 0.0, 0.3)} の 2 パターン、{(0.9, 0.5, 0.2, 0.2), ..., (0.9, 0.5, 0.2, 0.0)} の 3 パターン、および {(1.0, 0.4, 0.2, 0.3), ..., (1.0, 0.0, 0.2, 0.3)} の 5 パターン、{(1.0, 0.5, 0.1, 0.3), (1.0, 0.5, 0.0, 0.3)} の 2 パターン、{(1.0, 0.5, 0.2, 0.2), ..., (1.0, 0.5, 0.2, 0.0)} の 3 パターンが登録され、合計 $3 + (5 + 2 + 3) \times 3 = 33$ パターンが科目区分 1 の正例の集合に登録される。

一方、科目区分 2 の負例の集合には、まず、{(0.8, 0.5, 0.2, 0.3), (0.8, 0.4, 0.2, 0.3), ..., (0.8, 0.0, 0.2, 0.3)} を登録し、この各々に対し、科目区分 1 を 0.8 から 1.0 まで増加させたもの、科目区分 3 を 0.2 から 1.0 まで増加させたもの、科目区分 4 を 0.3 から 1.0 まで増加させたものすべての組み合わせを適用したものを登録する。同様に、科目区分 3 および科目区分 4 の負例の集合も作成する。

4.3 経験強化型学習利用下での正例および負例の集合を考慮した ACCS

4.3.1 正例および負例集合の活用方法

4.2.2 節で構築した正例および負例の集合を活用した初出科目の分類支援を考える。

まず, ある申請者が申告した科目中のすべての初出科目を ACCS へ入力し, 閾値パターンを作成する。得られた閾値パターンと4.2.2節で構築した正例および負例の集合とのマッチングをとることで, 各科目が, 各科目区分番号において, 正例, 負例, いずれの集合に属するかがわかる。

ACCS with XoL を用いることで, 各時点, すなわち, 各科目区分の必要単位数の満たされ具合に応じて, 満たすべき科目区分番号を知ることができる。そのため, ACCS with XoL が指定した科目区分であると判定される可能性が高い科目を専門委員に提示することが望ましい。

理想的には, 指定された科目区分において, 「負例の集合に含まれず, かつ, 正例の集合に含まれる科目」を専門委員に提示することができればよい。しかし, あるひとりの申請者が潤沢な科目数を持って申請することは稀であるため, そのような「負例の集合に含まれず, かつ, 正例の集合に含まれる科目」がつねに存在するとは限らない。

そこで, 本論文では, 負例および正例の集合との基本的な包含関係として, 次の PFP0 から PFP3 を考え, 評価実験により, それらの効果を比較する。

- PFP0: 負例の集合に含まれず, かつ, 正例の集合に含まれる科目集合
- PFP1: 負例の集合に含まれない科目集合
- PFP2: 正例の集合に含まれる科目集合
- PFP3: 正例の集合に含まれる, または, 負例の集合に含まれない科目集合

これらの間には, 「PFP0 \subset PFP1, PFP2 \subset PFP3」の関係がある。

4.3.2 正例および負例集合を活用した ACCS with XoL の全体像

ACCS with XoL に4.3.1節で述べた PFP k ($k = 0, 1, 2, 3$) を組み込んだ手法を ACCS with XoL/PN と呼ぶ。

一般には, 動的に, 使用する PFP を変更することも考えられる。しかし, ここでは, 各 PFP の違

いを適切に評価するため, 予め, 使用する PFP を固定しておく。以下に, その場合における, ACCS with XoL/PN の全体の流れを示す。

経験強化型学習利用下での正例および負例の集合を考慮した ACCS

step 0-0: PFP0, PFP1, PFP2, PFP3 の中からひとつの PFP k ($k = 0, 1, 2, 3$) を選択する。

step 0-1: 申請者が申告した科目のすべてを ACCS で評価し, 閾値パターンを作成する。

step 0-2: step 0-1 で作成された閾値パターンをもとに, 各科目が, 各科目区分番号において, 正例, 負例, いずれの集合に属するかを判定する。

step1: XoL を用いて, 審査の対象とする科目区分番号を決定する。(この処理は ACCS with XoL と同様である)

step2: PFP k に含まれる科目のうち, step 1 で決定された科目区分番号に対応する科目をひとつ選択し, 専門委員に判定していただく。(複数の科目が選択可能な場合は, 一様ランダムな確率でひとつの科目を選択する)

step3: 判定済みの科目を PFP k から除外するとともに, 修得単位の審査の基準の満たされ具合 (現在対象としている科目区分の必要単位数の満たされ具合) を更新する。

step 4: 修得単位の審査の基準が満たされていれば終了, 満たされていない場合は step 1 へ戻る。

5. 評価実験

本章では, 情報工学区分を対象に, 本論文で提案した ACCS with XoL/PN の性能評価を行う。

5.1 学習方法

ACCS with XoL/PN では学習データ, すなわち, 既に分類結果が既知である科目集合が必要となる。ここでは, 平成15および16年度に情報工学区分に実際に申請のあった4大学 (M, K, O, H) の各科目 (48, 28, 56および30科目) を学習データとして利用した。これは申請者4名分のデータである。なお, 修得単位の審査の基準を満たすために必要な最少科目数は, 各々, 8, 8, 25, 13科目である。

これらの学習データを用いて, 各申請者ごとに, 50万回修得単位の審査の基準が満たされるまで学

習を行った。学習時の行動選択は一様ランダムな確率でひとつの科目区分を選択するランダム選択を採用した。報酬は、文献 [16, 17] 同様、最少の科目提示数で修得単位の審査の基準が満たされたときのみ与えた。学習器 (XoL) としては、文献 [16, 17] で、修得単位の審査の基準を満たすために要した科目数が最も少なかった合理的政策形成アルゴリズム (Rational Policy Making algorithm; RPM) [10] を用いた。

また、同じ4大学の各科目を ACCS へ入力し閾値パターンを生成することで、各 PFP k ($k = 0, 1, 2, 3$) を構成した。

5.2 学習結果

評価データとしては、平成17年以降に申請のあった7名分 (申請者 A ~ G) のデータを用いた。これらは、主として大学で履修した科目を中心に申請してきた者たちのデータであり、科目数は、それぞれ40, 67, 27, 31, 27, 18, 49である。

RPM が、基準を満たすために要した科目の審査数を表3最上段に示した。実験はA~Gの各々に対し、乱数¹の種を変え、それぞれ100回行った。具体的には、その各100回の実験で先の4大学分 (M, K, O, H 大学) のデータを各々学習させ、その学習結果を用いて、A~Gまでの7名分のデータを100回学習させた²。表には、そのときの平均、ならびに括弧内には標準偏差を示した。さらに各 PFP の結果を、同じ形式で、その下に示した。

5.3 考察

PFP0 は、申請者 F で RPM と同等、それ以外はすべて改善している。しかし、本問題のように専門委員に判定していただくことが前提となってい

る場合には、依頼する科目数が極端に多いケースを減らすことが重要であると考えられる。このような視点で、各 PFP をみると、申請者ごとに大きく傾向が異なっている。特に科目数が最も少ない F では、負例の集合により科目候補数を絞る PFP (PFP0, PFP1) で改善がみられない。このように科目数が少ないケースでは、積極的に正例の集合を活用する PFP2 が好成績をおさめたものと考えられる。一方、科目数が多い B 等では、負例の集合により科目候補数を絞った方が若干好成績をおさめている。

C と E は同じ科目数 (27科目) であるが、それらの間では傾向が異なる。実は、E は27科目中6科目しか科目区分1~3に分類される科目が存在しない。そのため、科目数が少ない F に近い傾向になっていると考えられる。一方、C ではそのような極端なことは生じておらず、負例の集合による科目候補数の絞り込みが有効に機能していると考えられる。

各 PFP の中で、科目候補数が最も多いのは、PFP3 である。しかし、あまりに広げすぎて、E や F の場合でも、正解の科目区分には属さない可能性の高い科目を提示することが多くなり、PFP2 のような好成績はおさめられていない。

なお、試みに、まず、PFP2 で科目候補集合を作成し、後に (PFP2 による科目候補集合が空集合になった時点で)、PFP3 に移行する手法も行った。結果は、若干の改善はみられたが、ほぼ PFP2 と同一であった。

本問題は、申請者が履修した科目がもとになっているが、一般には、潤沢な科目数を持って申請することは希である。そのため、負例の集合による科目候補数の削減は、E や F の例のように逆効

表3 各学習手法の結果

	A	B	C	D	E	F	G
RPM	6.98 (0.07)	22.00 (0.00)	5.01 (0.04)	6.03 (0.23)	16.33 (0.36)	15.00 (0.00)	6.05 (0.14)
PFP0	4.41 (0.32)	4.40 (0.30)	4.40 (0.30)	5.40 (0.30)	15.40 (0.27)	15.00 (0.00)	4.00 (0.04)
PFP1	6.01 (0.09)	5.40 (0.30)	6.15 (0.08)	6.00 (0.00)	17.65 (0.29)	16.00 (0.00)	7.01 (0.11)
PFP2	4.75 (0.21)	6.37 (1.95)	6.55 (0.37)	7.00 (0.00)	9.66 (0.27)	4.67 (0.08)	7.41 (0.32)
PFP3	5.01 (0.09)	5.00 (0.00)	6.15 (0.08)	6.00 (0.00)	16.54 (0.25)	16.00 (0.00)	6.01 (0.10)

¹ 乱数発生アルゴリズムは、文献 [24] 記載のものを利用した。

² すなわち、A~Gの各申請者は、それぞれ1万回ずつ評価されることを意味する。

果になる場合が多い。潤沢か否かの判定は、専攻の区分によって異なると思われるが、情報工学区分の場合、およそ30科目以上の科目を申請していれば、PFP0のように負例の集合を積極的に利用し、科目候補数を減らすことが有効に機能する可能性が高いが、それより少ない申請数の場合は、PFP2のように積極的に正例の集合を考慮した方が、極端に提示する科目数が多くなるケースを抑えるという意味で、専門委員への負担軽減に寄与するものと考えられる。

6. おわりに

本研究ではNIAD-UEにおける科目分類支援システムの検討を行った。以前提案した経験強化型学習を用いた科目分類支援システム (ACCS with XoL) により適した形での科目分類支援システムの利用形態を提案し、情報工学区分を対象に有効性を確認した。

今後は、実運用可能なシステムを早急に開発したいと考えている。それには、情報工学区分以外の専攻の区分での検証が急務であると考えられる。しかし、現在基本としているACCSは、myDB_cの存在が前提となっており、現状では、それが整備されていない情報工学区分以外の専攻の区分での検証は行えない状況にある。そこで、今後は、文献 [14] で提案したような、初期には「専門科目名のリスト」を用いた分類支援を行い、判定結果が蓄積されるに連れ徐々にmyDB (myDB_c) を構築していく接近法が有望であると考えており、文献 [14] の手法を発展させる研究を進めているところである。

謝辞

本研究を遂行するにあたりデータ整理に多大なご協力を頂きました浦静香氏に謝意を表します。

参考文献

- [1] Chrisman, L.: Reinforcement Learning with perceptual aliasing: The Perceptual Distinctions Approach, Proc. of the 10th National Conference on Artificial Intelligence, 183/188 (1992)
- [2] 泉信一, 近藤基吉, 穂刈四三二, 永倉俊充編集: 共立数学公式附函数表改定増補, 共立出版株式会社, pp.439 (1969).
- [3] 井田正明, 野澤孝之, 芳鐘冬樹, 宮崎和光, 喜多一: シラバスデータベースシステムの構築と専門教育課程の比較分析への応用, 大学評価・学位研究, 2, 85/96 (2005)
- [4] 井田正明: 大学評価と情報技術の活用, オペレーションズリサーチ, 54, 5, 277/282 (2009)
- [5] Ida, M.: Consideration on Sensitivity for Multiple Correspondence Analysis, The International Multi Conference of Engineers and Computer Scientists 2010, 560/565 (2010)
- [6] 伊東栄典, 松永吉広, 山田信太郎, 廣川佐千男: Web シラバスからのDB構成, 2003年度人工知能学会全国大会 (JSAI2003), 1D4-08 (2003)
- [7] 井下理: シラバスの意味と機能, 大学セミナーハウス (編), 大学力を創る: FDハンドブック, 62/72, 東信堂 (1999)
- [8] 「言選 Web」, <http://gensen.dl.itc.u-tokyo.ac.jp/gensenweb.html>
- [9] 平成25年度版 新しい学士への途, http://www.niad.ac.jp/n_gakui/shinseishiryu/no7_5_gakushiH25_3.pdf
- [10] 宮崎和光, 荒井幸代, 小林重信: POMDPs 環境下での決定的政策の学習, 人工知能学会誌, 14, 1, 148/156 (1999)
- [11] 宮崎和光, 井田正明, 芳鐘冬樹, 野澤孝之, 喜多一: 電子化されたシラバスに基づく学位授与事業のための科目分類支援システムの試作, 情報処理学会論文誌, 46, 3, 782/791 (2005)
- [12] 宮崎和光, 井田正明, 芳鐘冬樹, 野澤孝之, 喜多一: 分類候補数の能動的調整を可能にした学位授与事業のための科目分類支援システムの提案と評価, 知能と情報, 17, 5, 558/568 (2005)
- [13] 宮崎和光, 山村雅幸, 小林重信: 強化学習における報酬割当ての理論的考察, 人工知能学会誌, 9, 4, 580/587 (1994)
- [14] 宮崎和光, 井田正明, 芳鐘冬樹, 野澤孝之, 喜多一: 専門科目名のリストを利用した学位授与事業のための科目分類支援システムの評価, 大学評価・学位研究, 6, 27/42 (2007)
- [15] Miyazaki, K. & Kobayashi, S.: Exploitation-oriented Learning PS-r#, Journal of Advanced

- Computational Intelligence and Intelligent Informatics, 13, 6, 624/630 (2009)
- [16] 宮崎和光, 井田正明: 経験強化型学習を利用した学位授与事業のための科目分類支援システムの提案, 第38回知能システムシンポジウム, 123/128 (2011)
- [17] Miyazaki, K. & Ida, M.: Proposal and Evaluation of the Active Course Classification Support System with Exploitation-oriented Learning, The 9th European Workshop on Reinforcement Learning (EWRL-9), Sept. 9, 2011, Athens Royal Olympic Hotel, Lecture Notes in Computer Science, 7188, 333/344 (2012)
- [18] 三好善彦: シラバスから行うカリキュラム分析, 埼玉女子短期大学研究紀要第25, 11/25 (2012)
- [19] 中川裕志, 森辰則, 湯本紘彰: 出現頻度と接続頻度に基づく専門用語抽出, 自然言語処理, 10, 1, 27/45 (2003)
- [20] 中挟知延子他: シラバスにおけるテキストマイニングの試み, 私立大学情報教育協会平成21年度教育改革 IT 戦略大会 (2009)
- [21] 野澤孝之, 井田正明, 芳鐘冬樹, 宮崎和光, 喜多一: シラバス文書のクラスタリングに基づくカリキュラム分析システムの構築, 情報処理学会論文誌, 46, 1, 289/300 (2005)
- [22] 野澤孝之, 芳鐘冬樹, 井田正明, 宮崎和光, 洪井進, 喜多一, 川口昭彦: ビジネス・MOT・会計, 公共政策系専門職大学院のカリキュラム構成—シラバスの文書クラスタリングを用いた比較分析—, 大学評価・学位研究, 5, 35/54 (2007)
- [23] 大場善次郎: 「工学知の構造化と可視化」の試み—工学教育に向けて—, 大学評価・学位研究, 1, 97/109 (2005)
- [24] Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T.: Numerical Recipes in C: The Art of Scientific Computing, Second Edition, Cambridge University Press (1992)
- [25] 東京大学工学部: シラバス構造化システム <http://www.t.u-tokyo.ac.jp/epage/>
- [26] 山田信太郎, 松永吉広, 伊東栄典, 廣川佐千男: Web シラバス情報収集エージェントの試作, 電子情報通信学会和文論文誌 D-I, J86, 8, 566/574 (2003)
- [27] 横井正美, 横井健: 高専シラバスの分析, 平成20年度電子情報通信学会東京支部学生会研究発表会 (2009)
- [28] Resnick, P. & Varian, H.: Recommender Systems, introduction to special section of Communications of the ACM, March 1997, 40, 3 (1997)
- [29] 関谷貴之, 山口和紀: 講義データベース—講義間の関係を構造化したオンラインシラバスの開発—, 日本知能情報ファジィ学会誌, 17, 5, 536/546, (2005)
- [30] Sutton, R. S. & Barto, A. G.: Reinforcement Learning: An Introduction, A Bradford Book, MIT Press (1998)
- [31] 渡辺将尚, 絹川博之, 芳鐘冬樹, 井田正明, 野澤孝之, 喜多一: シラバス文書からの情報抽出支援システムの試作, 情報処理学会第67回全国大会講演論文集, 475/476 (2005)
- (受稿日 平成25年11月21日)
(受理日 平成26年2月19日)

[ABSTRACT]

Proposal of the Active Course Classification Support System with Positive and Negative Examples and Combining with Exploitation-oriented Learning

MIYAZAKI Kazuteru*, IDA Masaaki*

The National Institution for Academic Degrees and University Evaluation awards academic degrees based on the accumulation of academic course credits. As a part of the degree awarding process, course credits must be classified for designated disciplinary fields according to pre-determined criteria. The course classification process is carried out based on careful reading of course syllabuses by subcommittee members of the institution who are well versed in the disciplinary field. Thus far we have developed an Active Course Classification Support system (ACCS) that is aimed at supporting subcommittee members in the course classification process. Recently, we introduced a learning mechanism called Exploitation-oriented Learning (XoL) into ACCS. In this paper, we improve the classification capability of ACCS combined with XoL by generating appropriate sets of positive and negative examples. We provide a numerical example to clarify the effectiveness of this improvement in the information engineering sub-field.

* Research Department, National Institution for Academic Degrees and University Evaluation