卒業研究

「LDAトピックモデルと履修成績を用いた履修レコメンドシステム」

国際文化学部　国際文化学科

1686592c 宮崎仁弥

指導教員：村尾　元教授

副指導教員：康　敏教授

目次

1. はじめに
   1. 研究の背景と目的
   2. 本論文の構成
2. 関連研究

2-1.

1. 使用した技術

3-1.　TF-IDF

3-2.　LDAトピックモデル

3-3.　コサイン類似度

1. 提案手法

4-1.　シラバスデータの取得・整形

4-2.　トピックベクトル生成

4-3.　レコメンド方法

1. 評価実験

5-1.　評価実験方法

5-2.　実験結果

1. おわりに

6-1.　まとめ

6-2. 今後の課題・展望

1. 謝辞
2. はじめに
   1. 研究の背景と目的

現在大学生が受講できる授業の数は大変多くなっている。例えば、神戸大学国際人間科学部の2020年に開講された授業数は約1500件である。そのため、学生は自分の趣味・嗜好に合わせて履修することが可能になっており、授業選択の自由度が高くなっている。しかし授業が多様化した反面、履修計画を建てることは煩雑化した。数ある科目の中からシラバスを確認し、自分が興味を持てる授業なのかなどの判断をしながら履修する科目を探し出すことはなかなか時間がかかる。神戸大学国際人間科学部グローバル文化学科の学生の履修科目とその成績のデータをもとに、科目選択の効率化や自分の知らなかった得意・興味のある科目の発見を促すことが本研究の目的である。

* 1. 本論文の構成

　本論文の構成は次のようになっている。第2章では、先行研究を紹介する。第3章では、本研究の提案システムの構成について記す。第4章では、今回用いた手法について述べる。第5章では、レコメンドシステムの評価の方法とその結果について述べる。第6章ではまとめと今後の課題・展望について述べる。

1. 関連研究

本研究と同様に大学生を対象に科目を推薦するシステムに関する既存研究が存在する。竹森ら[1]は、学部新入生を対象に教養科目を推薦するシステムの設計を行った。各科目の特徴をdoc2vecを用いてベクトル化し、その科目ベクトルに対してウォード法を用いたクラスタリングを行った。その次に高校主要科目の5科目ごとに科目ベクトルを作成する。各クラスタに属する各科目に対し、高校科目ベクトルとの類似度を計算し0~5の値で正規化し、ラーダーチャートに表し可視化する。大学の科目のクラスタのキーワードを可視化したワードクラウドを見て、学生はクラスタを選択し、そのクラスタの中の科目から学生はレーダーチャートも参考にしながら科目を選択する。科目をクラスタリングしているという点で共通しているが、本研究では成績を用いているという点で異なる。

西森ら[2]はTF-IDFとコサイン類似度を用いて科目間の類似度を求めた。履修する科目と履修済みの科目の類似度に直接GPAをかけることで、科目の成績を推定する。履修済みの科目に対して、推定を行ったところ、無作為に推定した場合より絶対平均誤差が低いことを明らかにした。本研究では類似度に直接GPAを掛け合わせるのではなく、トピックベクトルにGPAを掛け合わせている

1. 提案システムの構成

3-1.　全体の手順

3-1-1.　準備

本システムの構成は以下のようになっている。神戸大学国際人間科学部のシラバスの授業テーマのテキストデータからをTF-IDFを用いてベクトル化し、LDAトピックモデルを生成、科目のトピックベクトルを算出する。学生の履修履歴から「科目名」、「成績」、「時間割コード」、「開講年度」を取得する。

3-2-2.　レコメンド

　学生の成績データ内にある授業の年度と時間割コードによりシラバスファイルから検索し、該当する授業を探し出す。発見した授業のトピックベクトルそれぞれの値に成績データの値をかけ重みづけを行う。各授業のトピックベクトルに対してこの計算を行い、トピックごとに値を合計する。このようにして合計されたベクトルは学生の各トピックに対する嗜好性を表す。例えば、トピック1の値が大きければ、トピック1は得意だと考えられ、値が小さければ不得意であると考えられる。授業のトピックベクトルと学生の嗜好性ベクトルの類似度をコサイン類似度を用いて求める。コサイン類似度が大きいほど嗜好性に合っているため、その学生におすすめである。コサイン類似度が大きい授業から降順に並べ、上位の授業を学生にレコメンドする。

3-2. 個別の手法

3-2-1.　TF-IDF

TF-IDFは、文書内に出現する単語についてTF（出現頻度）とIDF（逆文書頻度）からその単語の重要度を求める手法である。TFとはTerm Frequencyの略である。これは各文書での単語の出現頻度を意味する。関数fを出現頻度を求める関数とし、文書における単語のTFを表したものが(1)式である。

(1)

しかし、TFのみではどの文書にも現れる単語の値も大きくなってしまう。そのような単語は文書の特徴を表しているとは考えにくい。そこで用いられるのがIDFである。IDFはInverse Document Frequencyの略である。これはある単語が含まれる文書の割合の逆数を表す。その単語の出現する文書の数が少ないほどこの値は大きくなる。ある文書集合における単語について考える場合、を単語が出現する文書数とすると、IDFは(2)式から求められる。

TF-IDFはTF値とIDF値を掛け合わせる以下の(2)式で求められる。

それにより、ある文書での出現回数は多いが、他の文書にはあまり出現しない単語のTF-IDF値は大きくなる。TF-IDF値が大きい単語ほどその文書の特徴を表していると言える。

3-2-2.　LDAトピックモデル

　LDAトピックモデルは、文書の確率的生成モデルとして提案された。LDAでは一つの文書に複数のトピックが存在すると仮定し、そのトピックの分布を離散分布としてモデル化する[3]。本研究ではシラバスの各授業のトピックの分布をLDAを使って求めている。表1は科目とそのトピック分布の一例である。ここではトピック数は6としている。トピック分布の例の数値は、各科目が6つのトピックにどれくらい属しているかを表している。現代IT入門Aはトピック5の値が一番大きいため、トピック5に最も属していると考えられ、グローバル社会動態発展演習Aはトピック4以外の値が0であるため、トピック4にのみ属していると考えられる。本研究ではこのトピック分布をトピックベクトルとみなし計算を行う。

表1.　科目とトピック分布の一例（小数点以下4位以下は切り捨て）

|  |  |
| --- | --- |
| 科目名 | トピック分布 |
| 現代IT入門A | 0.041, 0, 0.179, 0.101, 0.672, 0 |
| グローバル社会動態発展演習Ａ | 0, 0, 0, 0.977, 0, 0 |
| 国際関係論A | 0, 0.0323, 0.448, 0, 0.509, 0 |

なお、LDAトピックモデルでは自動的にトピック数を決定できないので手動でトピック数を決定する必要がある。その際の指標となるのがPerplexityとCoherenceである。

3-2-3.　Perplexity

Perplexityは平均分岐数とも訳され、トピックモデルによる予測精度を表している。Perplexityは小さいほどそのモデルの性能が良いことを示す。

3-2-4.　Coherence

　Coherenceは、トピックの質を表す。意味の近い単語が集まっているトピックをより多く抽出できる手法が良いモデルであるという観点からCoherenceの研究は行われている。トピックの質はトピック中の単語感類似度の平均値から求められる、トピック全体のCoherenceが高ければ良いモデルである。

3-2-5.　コサイン類似度

コサイン類似度はベクトル空間において、2本のベクトルがなす角度を表す指標である。以下の(4)式で求められる。1に近ければ類似しており、0に近ければ似てないことを表す。

(4)

1. 提案手法

4-1.　シラバスデータの取得・整形

　分析対象として神戸大学外部公開用シラバスを用いた。その中でも国際人間科学部の2016年から2020年の3,501授業の中から「科目名」、「時間割コード」、「開講年度」、「授業のテーマ」をPythonパッケージのBeautiful SoupとSeleniumを用いてスクレイピングした。授業のテーマのテキストデータにはJanomeを用いて形態素解析を行い、わかち書きをした。名詞が授業の特徴を表すと仮定し、わかち書きされたシラバスの単語群の中から名詞のみを抽出した。表2は授業名とそのシラバスの授業テーマから抽出された名詞の一例である。。

表2.　シラバスの一例

|  |  |
| --- | --- |
| 授業名 | 名詞 |
| 音楽文化史１ | エポックメーキング,音楽,作品,作曲,家,音楽,芸術,表現,様式,変遷,社会,文化,史,意味,考察 |
| 現代社会理論Ａ | 貧困,共有,事態,人類,歴史,共同,性,基礎,近代,後,個人,化,過程,貧困,忘却,進展,現代,私,自己,認識,社会,帰結,私,キーワード,現代,時,空間,認識 |
| 情報リテラシー演習１ | オンライン,コミュニケーション,文書,処理,計算,基本,操作,方法,身,情報,機器,具体,活用,技能,習得 |

4-2.　トピックベクトル生成

　トピックモデルの分析には、Pythonライブラリのgensimを用いた。LDAのトピック数は自動的には決まらないため、事前に指定する必要がある。本研究では、トピック数を決める際の指標として、PerplexityとCoherenceの2つを用いた。シラバスデータに対してトピック数を2~50に変化させ、PerplexityとCoherenceを求めプロットしたものを図2に示す。Perplexityは大きいほど良く、Coherenceは小さいほど良いとされるのでトピック数を6に設定した。トピック数6 にしてLDAを実行し、トピックに属する単語をワードクラウドで表示したものを図2に示す。ワードクラウドは各トピックにおける発生頻度が上位30位の単語を元に生成されており、発生頻度が大きいほど文字が大きくなっている。生成したLDAモデルを用いて、各授業のトピックベクトルを求める。

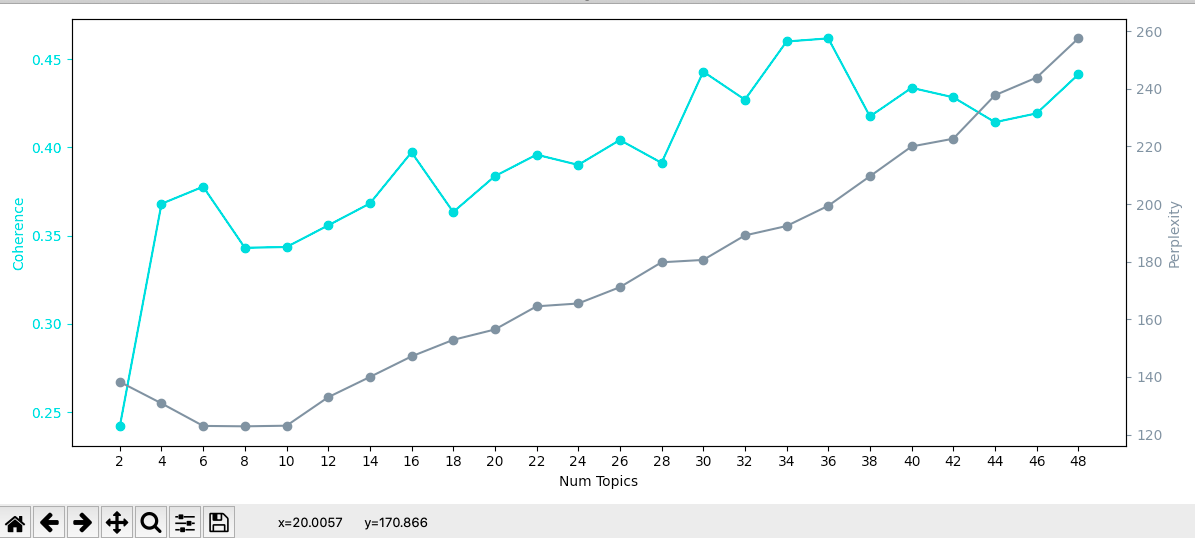


図 1.　PerplexityとCoherenceのプロット結果

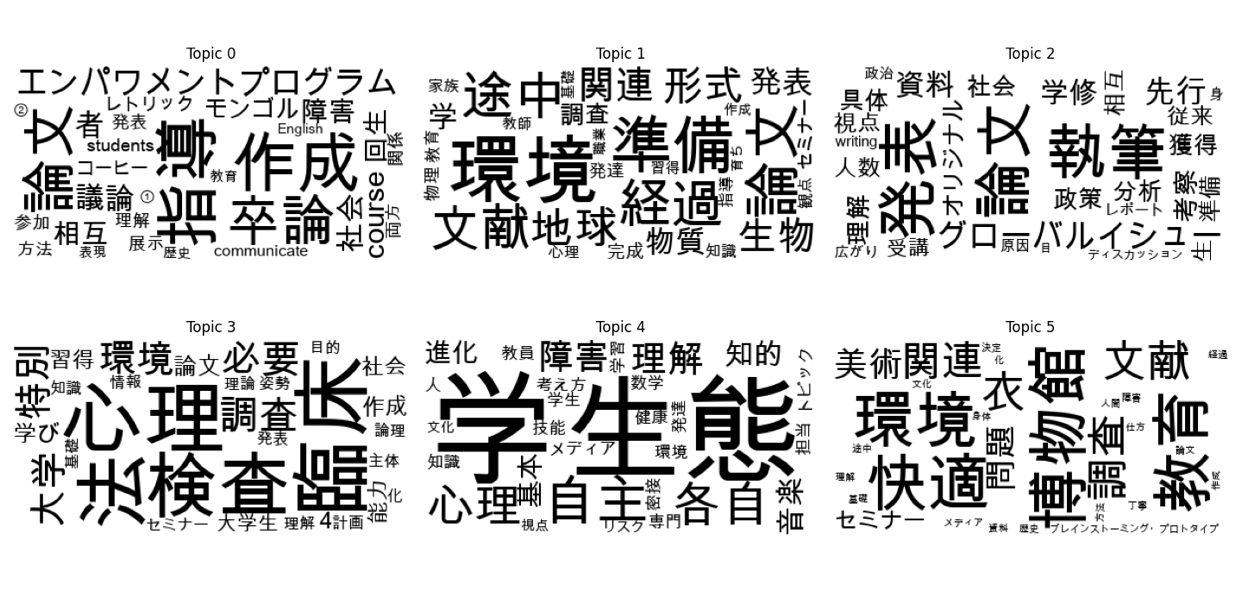


図2.　LDAによって生成された各トピックに属する単語をワードクラウドで表示したもの

4-3. レコメンド方法

4-3-1. 成績データ

提案するレコメンド方法では、シラバスデータと学生の成績データを用いる。学生の成績データは、うりぼーネットの履修成績照会にあるファイルを出力するボタンによって得られるCSVファイルを用いる。CSVファイルの成績の列は、GPAに合わせて、秀を4.2、優を4、良を3、可を2、不可を0に変換した。成績が「合格」となっている授業は基本的に必修授業のため、今回はおすすめに関与しないように0とした。履修取り消しされた授業は成績データから除いた。

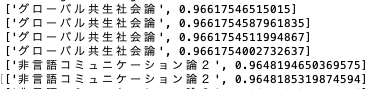
4-3-2.　嗜好性ベクトルの計算

学生の成績データ内にある授業の年度と時間割コードにより、該当する授業を探し、その授業のトピックベクトルそれぞれの値に成績データの値をかけ重みづけを行う。各授業についてこの計算を行い、トピックごとに値を合計する。このようにして合計されたベクトルは学生の各トピックに対する嗜好性を表す。例えば、トピック1の値が大きければ、トピック1は得意だと考えられ、値が小さければ不得意であると考えられる。

4-3-3. レコメンド

授業のトピックベクトルと学生の嗜好性ベクトルの類似度をコサイン類似度を用いて求める。コサイン類似度が大きい授業ほど嗜好性に合っているため、その学生におすすめである。コサイン類似度が大きい授業から降順に並べ、上位の授業を学生にレコメンドする。実際にターミナル常に出力された授業の例が図3である。

図3.　ターミナルに出力されたレコメンドの一例（右の値はコサイン類似度）



1. 評価実験

5-1.　評価実験方法

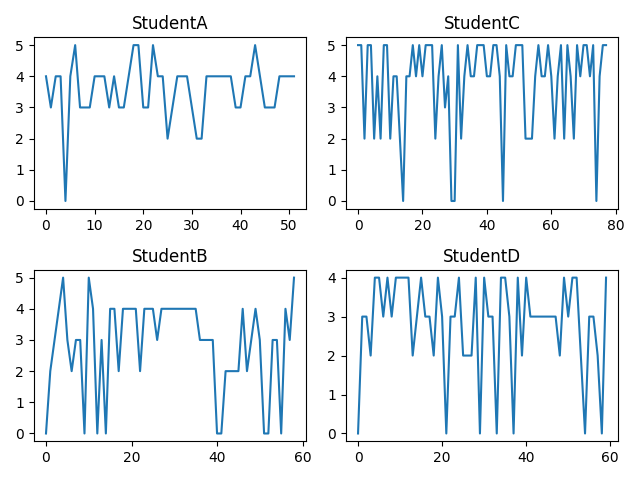
レコメンド方法についての評価にあたっては、成績評価のある既履修の科目の中からレコメンドし、レコメンドされた順番と成績の相関係数を見る方法を行った。成績とレコメンド順に正の相関関係があれば、学生の科目に対する得意・不得意に即したレコメンドができていると考えられる。実際のデータは、神戸大学国際人間科学部グローバル文化学部に所属する学生4名の履修成績データを用いた。

5-2.　実験結果

結果を表3に示す。最も相関係数が大きい学生の値でも約0.12であるためほとんど相関はない。横軸をレコメンドの順位、縦軸を成績としたプロットを図4に示す。レコメンドが適切であるならば、レコメンド順位が高い横軸の左部分に好成績の授業が、おすすめ度が低い右部分に成績の低い授業がプロットされ、左上から右下がりの図になる。しかし図を見てわかるように、好成績の授業は左右に散らばっており、反対に成績の低い授業も点在している。このことから、今回の手法が、学生の成績の良い科目をうまくレコメンドできているとはいえない。

表3.　成績とレコメンド順の相関係数

|  |  |
| --- | --- |
| 学生 | 相関係数 |
| 学生A | -0.0994319 |
| 学生B | 0.08027 |
| 学生C | -0.0578176 |
| 学生D | 0.116057 |

図4. プロット結果（横軸：レコメンド順, 縦軸：成績）

5-3.　考察

5-3-1. シラバスデータ

成績とレコメンド順に相関がない理由として大きく分けて3つの原因が考えられる。シラバスのテキストデータ、レコメンド方法、トピック以外の成績に影響する要因である。それぞれに対して考察していく。

　今回使用したシラバスの授業テーマのテキストデータは授業によってその書き方や量にばらつきがある。例えば近現代社会思想論Aの授業テーマは「近代社会をめぐる諸理論」のみであり、LDAモデル生成の際に用いられている単語は「近代、社会、理論」の3単語のみである。この3単語から授業の特徴を捉えることは難しい。このようにシラバスの文章量が少なく、特徴を捉えられていない授業があるためおすすめの精度が低下していることが考えられる。

5-3-2.　レコメンド方法

今回のレコメンド方法は授業のトピックベクトルに直接成績をかけて、トピックごとに合計して得られた嗜好性ベクトルと授業のトピックベクトルのコサイン距離が小さいものをレコメンドするという方法である。しかし、この方法以外にも学生が一番得意なトピックの値が大きい授業をおすすめする方法や他の学生の嗜好性ベクトルを用いて協調フィルタリング的におすすめする方法もなども考えられる。

5-3-3.　 トピック以外の成績に影響する要因

今回はトピックベクトルと成績のみを用いてレコメンドを行った。図5は得られた嗜好性ベクトルとレコメンドされた授業のトピックベクトルをターミナルに出力したものである。上に表示されてるほどレコメンド順が高い。これを見てみるとこの学生はトピック4の値が約46.6と最も大きくおすすめされている授業も全てトピック4の値が最も大きい。今図6は授業のトピックベクトルと成績を表示したものである。認知コミュニケーション論1, 2、近現代社会思想論A, Bはそれぞれトピックベクトルが非常に近い。しかし学生の成績はそれぞれ2と5、4と3など異なっており、学生は授業のトピックベクトルが近い授業でも成績にばらつきがあることがわかる。この要因としては、学生の履修したときの状況や成績評価基準の相違などが考えられる。例えば、体調の悪化や学外の用事などがあれば、授業への参加率は下がり成績が下がる。その他には、同じテーマの授業でも成績の評価がテスト形式からレポート形式に変われば成績が異なる場合が考えられる。このように学生の成績には授業のテーマに現れない様々もあり、テーマのみから推定されるトピックを用いてレコメンドした場合、必ずしも成績が良いもののレコメンド順位が高くならないということが考えられる。

図5.　嗜好性ベクトルとおすすめされた授業のトピックベクトルの一例

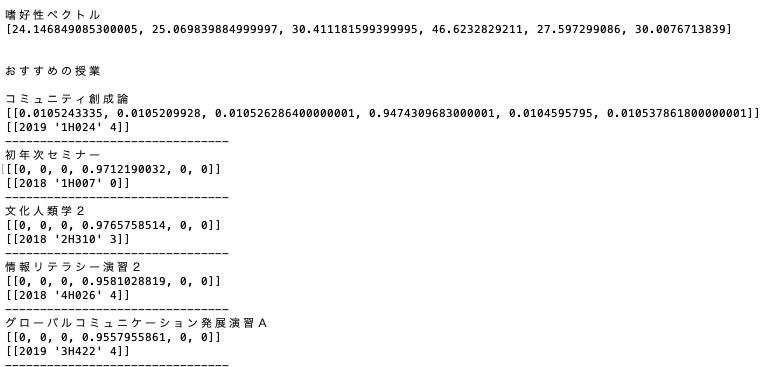
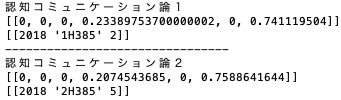
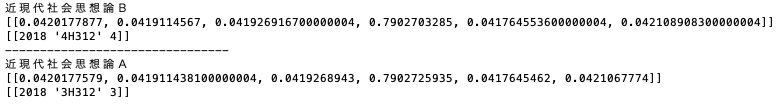


図6. 授業のトピックベクトルと成績





1. おわりに

6-1.　まとめ

本研究では、LDAトピックモデルと成績を用いた履修レコメンドシステムを開発し、成績とレコメンド順の相関を求めることでその評価を行った。その結果、成績とレコメンド順には相関関係が見られず、今回のレコメンド方法では適切なレコメンドができていないことが確認された。

6-2.　今後の課題と展望

レコメンドがうまくいかなかった原因として、データ、レコメンド方法、トピック以外の要因の成績に対する影響を挙げた。データの問題は、一部の授業のデータの量が少なかったことに起因すると考えられる。そこで今回用いた授業テーマだけでなく、授業の概要と計画や成績の評価方法などを用いることによって、データの量と多様性を増やすことで解決できる可能性がある。レコメンド方法は、コサイン類似度以外にも、一番得意なトピックのみを考慮しておすすめする方法が考えられる。トピック以外の成績に影響する要因は、成績評価方法や学生の環境の変化などが考えられる。前者はデータの問題でも述べたようなシラバスの成績評価基準を用いることで考慮に入れることができる。後者は要因を特定することが難しい。履修成績にある授業に対して持っていたモチベーションをアンケートによって集め、モチベーションと成績の観点から分析することができるかもしれない。

トピックと成績の相関は今回の研究では見られなかったが、トピックの精度をあげることや多様なデータと組み合せることで、LDAトピックモデルと履修成績を用いた授業レコメンドができる可能性がある。また、今後はLDAトピックモデルに加えて。協調フィルタリングの技術を用いて、複数の学生の履修成績を考慮に入れた履修レコメンドシステムの研究を行っていきたい。

参考文献

[1] 竹森汰智, 亀井清華. “科目推薦のためのdoc2vecの応用方法の検討”. 情報処理学会. 2018

[2] 西森友省, 堀幸雄, 今井慈郎. “履修履歴を用いた科目推薦システム”. 情報処理学会. 2013.

[3] 奥村学, 佐藤一誠. “トピックモデルによる統計的潜在意味解析”. コロナ社. 2015.

LDAトピックモデルと履修成績を用いた

履修レコメンドシステム

所属：国際文化学部

学籍番号：1686592C

氏名：宮崎仁弥

本論文では「LDAトピックモデルと履修成績を用いた履修レコメンドシステム」について記す。

大学生が受講できる授業の数は大変多くなっている。そのため、学生は自分の趣味・嗜好に合わせて履修することが可能になっており、授業選択の自由度が高くなっている。しかし授業が多様化した反面、履修計画を建てることは煩雑化した。数ある科目の中からシラバスを確認し、自分が興味を持てる授業なのかなどの判断をしながら履修する科目を探し出すことはなかなか時間がかかる。そこで、履修スケジュールを考える時間の短縮と履修成績の分析による学生の得意・不得意なトピックの分析を目的として、本研究ではLDAトピックモデルと履修成績を用いた科目レコメンドシステムを構築、その評価を行った。

科目のシラバスデータは、神戸大学学外公開用シラバスの国際人間科学部の2016~2020年度のシラバスデータを用いた。

レコメンド方法は以下の通りである。学生の履修済みの授業のトピックベクトルに成績を数値化したものをかけ、トピックごとに合計したものを、その学生のトピックに対する嗜好性ベクトルとする。これから履修する科目の中からその科目のトピックベクトルと嗜好性ベクトルのコサイン類似度が高い科目をおすすめする。

このレコメンドシステムの評価を行った。評価には、履修成績の中から授業をレコメンドし、そのレコメンドされた順番と成績の相関関係を求める方法を用いた。その結果、レコメンド順と成績の間には相関がないことが明らかになった。その原因として、シラバスのテキストデータ、レコメンド方法、トピック以外の成績に影響する要因の3つが考えられる。今回用いたデータはシラバスの内の授業のテーマの部分のみだったので、科目によっては文量が非常に少ない科目もあり、トピックを適切に抽出できなかった可能性がある。レコメンド方法に関しては、得意なトピックのみに基づいてレコメンドする方法や協調フィルタリングを用いることで他の学生のデータを考慮に入れてレコメンドする方法が考えられる。また、学生の履修履歴のみではなく、履修した際の学生の心理的状況や環境などのトピック以外の成績に影響している要因を考慮することで、レコメンドの精度が上がることが考えられる。

付録

ソースコード

――――――――――――――――――――――――――――――――――――――――

test\_reccomendation.py

|  |
| --- |
| 1. import pandas as pd 2. import pickle 3. import csv 4. import json 5. import numpy as np 6. import scipy.stats 7. topic\_num = 6 8. sum\_topic\_odds = [0] \* topic\_num 9. def get\_topic\_value(nendo,code): 10. topic\_grades = None 11. taken\_class = df[(df['年度'] == nendo) & (df['時間割コード'] == code)] 12. topic\_value = taken\_class['トピックの確率'].values.tolist() 13. return topic\_value 14. def search\_goodat\_topic(nendo, code, grade): 15. topic\_grades = None 16. taken\_class = df[(df['年度'] == nendo) & (df['時間割コード'] == code)] 17. topic\_value = taken\_class['トピックの確率'].values.tolist() 18. if len(topic\_value) == 1: 19. topic\_grades = [n \* grade for n in topic\_value[0]] 20. return topic\_grades 21. def cos\_sim(v1, v2): 22. v1\_array = np.array(v1) 23. v2\_array = np.array(v2) 24. return np.dot(v1, v2) / (np.linalg.norm(v1) \* np.linalg.norm(v2)) 25. def reccomend(nendo,code): 26. \_class\_names\_cos\_sim = [] 27. taken\_class = df[(df['年度'] == nendo) & (df['時間割コード'] == code)] 28. class\_names\_topics = taken\_class[['科目名','年度','時間割コード','トピックの確率 29. ']].values.tolist() 30. for class\_name\_topic in class\_names\_topics: 31. similarity = cos\_sim((scipy.stats.zscore(class\_name\_topic[3])),sum\_topic\_odds) 32. class\_name\_topic[3] = similarity 33. \_class\_names\_cos\_sim.append(class\_name\_topic) 34. return \_class\_names\_cos\_sim 35. df = pd.read\_json('syllabus\_tfidf.json') 36. with open('data/grades/StudentA.csv') as f: 37. h = next(csv.reader(f)) 38. reader = csv.reader(f) 39. grades = [e for e in reader] 40. f.close() 41. df2 = pd.read\_csv('data/grades/StudentA.csv') 42. count = 0 43. for row in grades: 44. topic\_grades = search\_goodat\_topic(int(row[0]), row[1], float(row[2])) 45. if topic\_grades is not None: 46. sum\_topic\_odds = [topic\_grades[i] + sum\_topic\_odds[i] for i in range(len(topic\_grades))] 47. count += 1 48. class\_names\_cos\_sim = [] 49. for row in grades: 50. class\_names\_cos\_sim.extend(reccomend(int(row[0]), row[1])) 51. reccomend\_class = sorted(class\_names\_cos\_sim, reverse=True, key=lambda x: x[3]) 52. print("\n嗜好性ベクトル") 53. print(sum\_topic\_odds) 54. print('\n') 55. print("おすすめの授業\n") 56. for i in reccomend\_class[0:10]: 57. print(i[0]) 58. topic\_value = get\_topic\_value(i[1],i[2]) 59. print(topic\_value) 60. class\_grade = df2[(df2['年度'] == i[1]) & (df2['時間割コード'] == i[2])] 61. print(class\_grade.values)   print('--------------------------------') |