卒 業 研 究 報 告 題 目

文書の類似度計算における
次元削減手法に関する研究

A Study on Dimension Reduction Method in Similarity Calculation of Documents

指 導 教 員　　出口 利憲 教授

岐 阜 工 業 高 等 専 門 学 校　電気情報工学科

2014E26 長尾 彪真

平 成 ３ １ 年（ ２ ０ １ ９ 年 ）　　２ 月 １ ５ 日 提 出

Abstract

This study has two purposes. First purpose is calculating the similarity of syllabuses in the department of electrical and computer engineering in National Institute of Technology, Gifu College by text mining. If similarity of syllabuses becomes clear, you can grasp the relation between subjects. So you can study more effective. Second purpose is comparing the methods to reduce dimension. In this study, the new method is prepared in addition to conventional methods such as LSA and LDA. The new method uses cluster analysis and concept distance which calculated from Japanese WordNet. The merit of new method is considering the meaning of words. But, it is difficult to judge whether the new method is truly capable. Therefore, the capability is verified to compare with LSA and LDA. As a result, the new method is not capable compared with LSA(Figure 1). The main reason is shortage of Japanese WordNet's vocabulary. However, the new method has possibility to improve capability by using English WordNet.

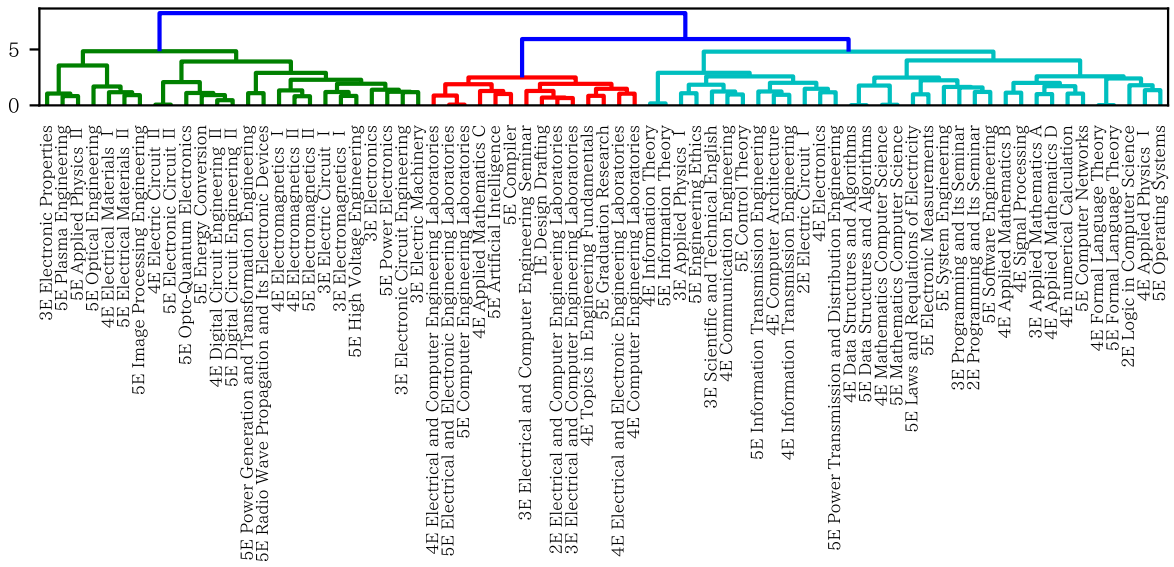


Figure 1　Result of dimension reduction by cluster analysis

目 次

第1章　序論

現代社会では、ITインフラが整備されたことにより、莫大な量のデータが発信、収集、記録されている。そのため、インターネットに接続すれば、様々な情報を簡単に入手することができる。しかし、中には虚偽の情報や信頼性の低い情報も存在し、人の手で本当に有益な情報を見つけ出すことは困難になりつつある。そこで、コンピュータを使って大量のデータから有益な情報を見つけ出す、データマイニングという技術が生まれた。その中でも、テキストデータを対象としたマイニングをテキストマイニングという。テキストマイニングは、テキストデータに出現する単語の頻度や傾向を解析し、データの相関関係や特徴を可視化する技術である。企業に導入される事例も増加傾向にあり、電話での問い合わせ内容の分類や、アンケートの自由記述欄に記述された感想、要望の傾向分析等に用いられている。

本研究では、テキストマイニングにより、本校電気情報工学科の科目のシラバスについて類似度計算を行った。シラバスの類似度からは科目間のつながりが分かり、ある科目で習得した知識や技術を、その他のどの科目で活用できるのかを把握することができる。これにより、高学年で受講する科目について、より効率的な学習ができる。

今回用いるシラバスのデータは、そのままでは次元が大きいため、類似度計算を行う前にデータの次元削減を行う。次元削減の手法には、従来から潜在的意味解析 (LSA : Latent Semantic Analysis) や潜在的ディリクレ配分法 (LDA : Latent Dirichlet Allocation) といったものが存在する。しかし、これらは数学的な手法であり、単語の持つ意味は考慮されない。そこで、日本語 wordnet とクラスター分析を組み合わせ、単語の意味を考慮した次元削減ができる手法を考えた。この手法と従来の手法で次元削減をした結果を比較し、有効性について検討した。

第2章　テキストマイニング

2.1　テキストマイニング

2.1.1　データマイニング

データマイニングは、データベースに蓄積された大量のデータを解析することで、有益な情報を見つけ出す技術である。対象となるデータは、飲食店で言えばメニューごとの売り上げ個数や来店人数、客層といったものであり、量が多いというだけでなく種類も複数存在する。そのため、これらのデータから、来店傾向のような法則性を手動で見つけ出すのは困難である。そこで、コンピュータを使用することにより、高速かつ低コストでデータの法則性や相関を発見する必要がある。このような場面で、データマイニングが用いられる。データマイニングには様々なアプローチがあり、その一つに本研究で使用するクラスター分析がある。

2.1.2　テキストマイニング

テキストマイニングは、テキストデータを対象としたデータマイニングのことである。言葉は意味を持っているため、数値を対象としたマイニングと比較して、分析の難度が高い。

具体的な処理は二段階に分かれており、まずは文章を単語や文節に区切る処理を行う。その後、出現する単語の頻度や傾向を解析し、有益な情報を発見する。テキストマイニングが適用されている事例は序論で述べた通りで、主に問い合わせ内容やアンケート結果の解析である。本研究では、シラバスの類似度計算を行い、科目ごとの関連を可視化するためにテキストマイニングを用いた。

2.1.3　形態素

形態素とは、意味を持つ最小の単位で、それ以上分割できない言葉のことである。厳密には単語と異なっており、形態素がそのまま単語ととなる場合と、そのままでは単語にならない場合がある。

2.1.4　形態素解析

形態素解析は、テキストデータに含まれる文章を形態素に分割する手法である。この手法によって形態素に分割することで、それらの出現頻度等を類似度計算に利用することができる。形態素解析を行うエンジンは複数公開されており、本研究では MeCab を使用した。

2.1.5　MeCab

MeCab は、オープンソースの形態素解析エンジンである。日本語に対応しており、日本で使用されるツールの中で最もメジャーである。また、形態素の情報が記録された辞書も同じサイトに用意されているため、同時にインストールすることができる。MeCab によって形態素解析をするときは、オプションを指定することで様々な結果を出力できる。以下に、形態素とともに品詞、標準形等を出力した結果を示す。

すももももももものうち

すもも 名詞, 一般,*,*,*,*, すもも, スモモ, スモモ

も 助詞, 係助詞,*,*,*,*, も, モ, モ

もも 名詞, 一般,*,*,*,*, もも, モモ, モモ

も 助詞, 係助詞,*,*,*,*, も, モ, モ

もも 名詞, 一般,*,*,*,*, もも, モモ, モモ

の 助詞, 連体化,*,*,*,*, の, ノ, ノ

うち 名詞, 非自立, 副詞可能,*,*,*, うち, ウチ, ウチ

2.2　自然言語

2.2.1　自然言語

自然言語とは、人間が日常的に使用する、意思疎通を行うための言語である。それに対して、コンピュータで使用することを目的に作られたプログラミング言語等を人工言語という。両者にはもちろん文法が存在するが、言語の解釈の仕方に関して大きな違いがある。自然言語では、文法に則って記述された文章であっても、読み手によって複数通りの解釈がされる場合がある。また、主語が抜けているといった文法上の問題がある文章が、相手に伝わる場合もある。しかし、人工言語では、解釈は一通りしか存在せず、命令に対する処理は決まっている。その理由は、命令に対して複数の解釈があると、コ

ンピュータの処理が毎回変化してしまうからである。

2.2.2　自然言語の曖昧さ

自然言語の曖昧さには、後述する二つの性質がある。一つ目は多義性であり、単語に複数の意味が存在し、状況に応じて解釈が変わるという性質である。例えば、「適当」という単語には、「いい加減」と「ちょうどよい」という二つの意味が存在する。「適当な人材」という文であれば適当の意味は前者になるが、「適当な扱い」という文だと意味は後者になる。二つ目は類義性であり、異なる単語が同じ意味を持つという性質である。例えば、「雷」と「稲妻」では、読みも漢字も全く異なるが、ほぼ同じ意味を持っている。このような曖昧さがあるため、コンピュータによる自然言語処理は難度が高くなっている。

第3章　実験で使用した技術

3.1　単語の重要度と文書の類似度を算出する手法

3.1.1　TfIdf

TfIdf とは、各文書において単語が持つ重要度であり、Tf と Idf の積をとることで算出できる。Tf は Term Frequency の略であり、文書中の単語の出現頻度を表す。同じ単語が同じ文書内で多く出現するほど Tf は大きくなり、重要度も増す。Idf は Inverse Document Frequency の略であり、単語の情報量を表す。ある単語について、出現する文書数が多いほど Idf は小さくなり、文書数が少ないほど大きくなる。つまり、少数の文書のみで出現する単語の重要度を上げ、文書の特徴語を強調する効果がある。

文書数を N、N 個の文書に出現する単語の種類を M とおき、各文書を $d_i(i = 1, 2, \ldots, N)$、文書 d_i に出現する各単語を $t_{ij}(j = 1, 2, \ldots, M)$ とすると、Tf, Idf, TfIdf は次の式で求められる。

$$Tf_{ij} = \text{文書 } d_i \text{ における } t_{ij} \text{ の出現回数} \tag{3.1}$$

$$Idf_{ij} = \log \frac{\text{全文書数}}{\text{単語 } t_{ij} \text{ が出現する文書数}} \tag{3.2}$$

$$TfIdf_{ij} = Tf_{ij} \times Idf_{ij} \tag{3.3}$$

Idf については、全文書に出現する単語の値が 0 にならないよう、1 を足すことがある。しかし、全シラバスに出現する単語は類似度に影響がないと考えられたため、本研究では 1 を足す処理を行わなかった。また、Idf の計算式にある log の底は e とした。

3.1.2　cos 類似度

cos 類似度とは、ベクトル空間モデルにおいて、文書同士の類似度を計算する手法である。この手法により算出した値は、二つの文書ベクトルのなす角度の近さであり、文書同士の類似度を表している。三角関数の cos と同様で、値が 1 に近いほど二つの文書は類似しており、0 に近いほど類似していない。

ベクトル $\vec{a}$ とベクトル $\vec{b}$ の cos 類似度は、次の式で求められる。

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{3.4}$$

3.2　次元圧縮の手法

3.2.1　主成分分析 [1)]

実験において、解析対象となるデータが少ない場合は、グラフや統計量を用いてデータが持つ特性を容易に把握することができる。しかし、実際はデータの量や種類が多いため、関係が複雑になり、解析が困難になる場合がほとんどである。そのような問題を解決する手法として、主成分分析 (PCA : Principal Component Analysis) が存在する。主成分分析は、多次元のデータについて、本来持っている情報をできる限り損なうことなく次元を削減する手法である。この手法には、データの解析が容易になるだけでなく、視覚的にも理解しやすいという利点がある。

具体的な手順について、以下に式を用いて説明する。P 個のデータ $x_p(p = 1, 2, \ldots, P)$ について、$N(N \leq P)$ 個の主成分 $z_n(n = 1, 2, \ldots, N)$ とこれらの関係は、次の式のように互いに独立な線形結合として表される。

$$z_n = \sum_{p=1}^{P} a_{pn} x_p \tag{3.5}$$

ここで、z_n は第 n 主成分と呼ばれ、その結合係数 a_{pn} は次の式を満たす必要がある。

$$\sum_{p=1}^{P} a_{pn}^2 = 1 \qquad (\forall n) \tag{3.6}$$

主成分ができる限り多くの情報を持つようにするためには、データの分散に着目し、結合係数を上手く決める必要がある。例えとして、Figure 3.1 に示す二次元のデータについて考える。この図において、データの分散が最も大きくなる方向に着目すると、z_1 という軸ができる。これが第 1 主成分となり、このような軸ができるように式 (3.5) の結合係数を決定する。しかし、この軸だけでは、データが本来持つ情報を十分に表しているとは言い難い。そこで、z_1 に次いでデータの分散が大きくなる方向に着目し、z_2 という軸をとる。これが第 2 主成分となり、第 1 主成分にて表せないデータを補うことができる。このように結合係数を決めていくことで、情報量の損失を最小限に抑えながら、Figure 3.1 に示される X, Y の特性を把握することができる。今回の例では 2 次元の簡単なデータを取り扱ったため、主成分分析の利点が分かり辛い。しかし、高次元のデータになると、データ量だけでなく分散も大きくなり、主成分分析の利点がはっきりと表れるよう

になる。

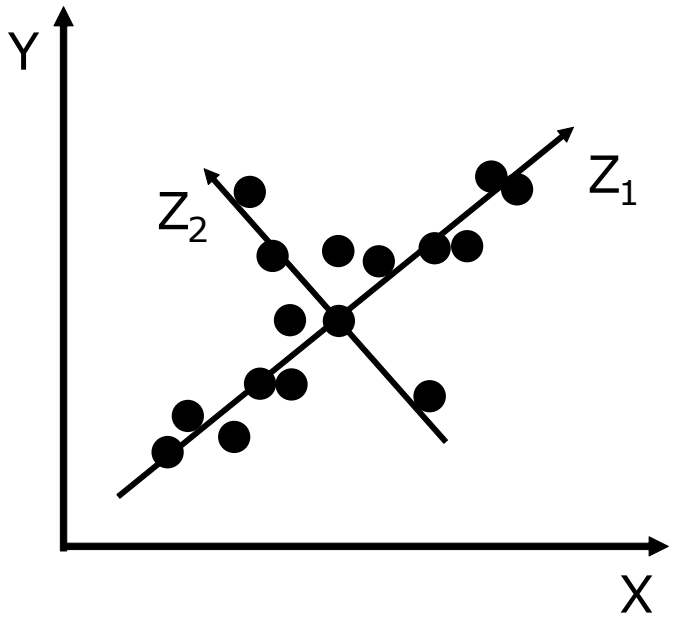


Figure 3.1　Example of two-dimensional data

主成分分析では、分析の対象となるデータ行列について、その共分散行列の固有値が、主成分の分散と等しくなる。ここで、各固有値は、大きいものから順に第 1 主成分, 第 2 主成分, . . . , 第 N 主成分に対応する。つまり、固有値が大きいほど、多くの情報を持っているということになる。

3.2.2　主成分数の設定

主成分分析においては、主成分数をいくつに設定するかが重要となる。もし主成分を少なくしすぎると、本来持つ情報が大きく損なわれてしまう。その一方で、多すぎても次元が削減されず、主成分分析が意味を成さない。そこで、主成分の分散と等しい共分散行列の固有値に着目し、以下に示す三通りの方法によって主成分数を設定する。

- 固有値が 1 を越える主成分を採用する。
- ある固有値とその次の固有値の差が小さくなるまでの主成分を採用する。
- 累積寄与率がある値に達するまでの主成分を採用する。

一つ目の方法は、平均と分散を共に 1 としたことで、分散がこの標準化された値である 1 よりも大きければ、説明力のある主成分として用いることができるという考えに基づいている。また、二つ目の方法は、ある固有値とその次の固有値の差が小さければ、

主成分の採用、非採用の区別に大きな意味はないという考えに基づいている。最後の方法については、主成分分析後のデータが、本来のデータが持つ情報の何割かを含んでいれば良いという考えに基づいている。この場合、累積寄与率が60〜80パーセントに達するまでの主成分数を採用することが多い。

累積寄与率とは、全ての主成分の寄与率を足し合わせた値である。また、寄与率というのは、ある主成分の固有値が表す情報が、全ての情報のうちどの程度の情報を表しているかを示す値であり、次式で表される。

$$P_n = \frac{\lambda_n}{\sum_{p=1}^{P} \lambda_p} \tag{3.7}$$

ここで、式中のλ_nは、n番目の主成分の固有値を示している。累積寄与率はこの寄与率の総和であるため、主成分数がNの場合は次式で表される。

$$C_n = \sum_{i=1}^{N} P_i \tag{3.8}$$

3.2.3 LSA[2)]

テキストマイニングにおいて、形態素解析後に生成される単語-文書行列は、高次元であることが多い。次元が高いと、文書の分類や特徴の抽出といった処理を行う際の計算量が多くなり、結果が出力されるまでに時間がかかる。また、分類の妨げとなる単語も多く存在し、それらがノイズになることもある。これらの問題を解決する時に用いられるのが、潜在的意味解析 (LSA : Latent Semantic Analysis) である。潜在的意味解析は、文書データから潜在的なトピックを推定し、そのトピックの数まで次元を削減する手法である。潜在的意味インデキシングとも呼ばれる。

LSA では、特異値分解という行列分解手法を用いて次元を削減する。以下に、文書行列TDを特異値分解する式を示す。

$$TD = U\Sigma V^T \tag{3.9}$$

この式におけるU, Σ, V^Tは行列を表しており、それぞれ左特異 (ターム) ベクトル、特異値を含むベクトル、右特異 (文書) ベクトルと呼ばれる。 特異値分解で得られた左特異ベクトルは左にある成分ほど重要度が高くなっているため、左からk列を抜き出した行列U_kを式 (3.10) のように掛け合わせることで、元の行列の持つ情報をできるだけ保った

まま次元を削減した行列を生成することができる。

$$TD_k = U_k^T TD \tag{3.10}$$

3.2.4 pLSA と LDA[2)3)]

pLSA は、Probabilistic Latent Semantic Analysis の略で、確率的潜在意味解析と呼ばれる。LSA に確率の概念を加えた手法で、各文書が各トピックに属する確率を算出する。pLSA では、すべての値が確率で表現されるため、負の値をとる、各単語が一つのトピックのみにしか属さないという仮定をしているといった LSA の問題を解消できる。具体的には、以下に示す処理の流れをたどる。ここで、各文書における各トピックの重要度割合をパラメータ 1、各トピックにおける各単語の重要度割合をパラメータ 2 とする。

1. パラメータ 1 とパラメータ 2 を適当に決定する。
2. パラメータ 1 を基に文書へのトピックの割り当てを行う。
3. パラメータ 2 を基にトピックへの単語の割り当てを行う。
4. 3 の処理をその文書に出現する全ての単語について行う。
5. パラメータ 1 とパラメータ 2 を更新する。
6. 2〜5 の処理を全文書について行う。

LDA は、Latent Dirichlet Allocation の略で、潜在的ディリクレ分布法と呼ばれる。pLSA をさらに発展させた手法で、ディリクレ分布を基に確率の計算を行う。ディリクレ分布とは、ある n 個の事象について、i 番目の事象が $\alpha_i - 1$ 回発生する場合の確率が x_i である確率を表す分布であり、次の式で表される。

$$p(x;\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{n} x_i^{\alpha_i - 1} \qquad B(\alpha) = \frac{\prod_{i=1}^{n} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{n} \alpha_i)} \tag{3.11}$$

ここで、この式においては、$x_i \geq 0$, $\Sigma x_i = 1$, $\alpha_i > 0$ である。また、x_i の期待値と分散については、次の式で表される。

$$E[x_i] = \frac{\alpha_i}{\sum_{j=1}^{n} \alpha_j} \tag{3.12}$$

$$var[x_i] = \frac{\alpha_i \sum_{j \neq i} \alpha_j}{(\sum_{j=1}^{n} \alpha_j)^2 (1 + \sum_{j=1}^{n} \alpha_j)} \tag{3.13}$$

LDA では、まず各文書におけるトピックの分布に関する事前分布と、各トピックにおける単語の分布に関する事前分布を仮定する。その後、それらの分布を繰り返し更新していくことで、各文書が各トピックに属する確率と各単語が各トピックに属する確率を算出する。pLSA と比較すると、汎化性能が高く、拡張しやすいという利点がある。言い換えると、確率が算出された状態で新しい文書が追加されても、その文書に関してトピックの分類が可能ということである。

3.2.5 クラスター分析

クラスター分析とは、対象となるデータを、似た性質を持つもの同士の集合に分類する手法である。教師なしの分類手法であり、データの中から分類の基準を見つけ出すことができる。ここで、この手法によって作成された集合を、クラスターと呼ぶ。

クラスター分析は、分類が階層的になる階層的クラスター分析と、あらかじめクラスター数を指定して分類する非階層的クラスター分析の二種類に分けられる。この二種類の内、本研究では階層的クラスター分析を利用する。階層的クラスター分析では、データ間の距離を基に最も距離の近いデータから順次結合していき、クラスターを形成していく。また、形成された階層構造は、Figure 3.2 に示すデンドログラムによって表される。デンドログラムにおいては、結合点が末端に近いほど距離が近い、すなわち類似度の高いデータといえる。階層的クラスター分析はクラスター間の距離の測定方法に複数種類が存在するが、その中でも最近隣法、最遠隣法、群平均法、ウォード法について説明する。

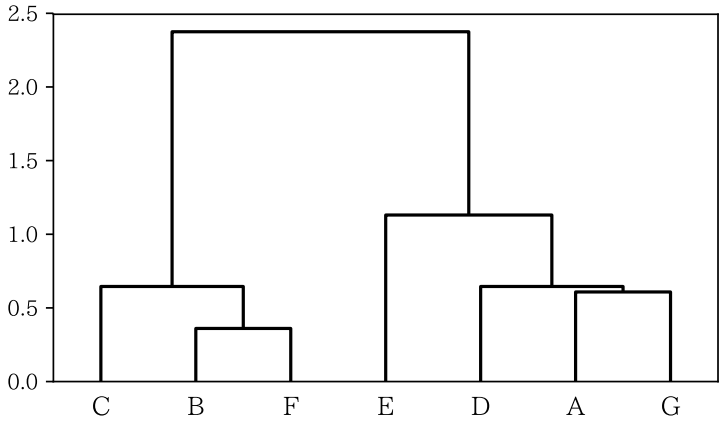


Figure 3.2　Dendrogram

最近隣法

2 つのクラスターの中で、最も近いデータ間の距離をクラスター間の距離とする方法。Figure 3.3(a) では、距離 $D_{(b,d)}$ をクラスター A とクラスター B の距離として、Figure 3.3(b) のクラスターを形成する。

最遠隣法

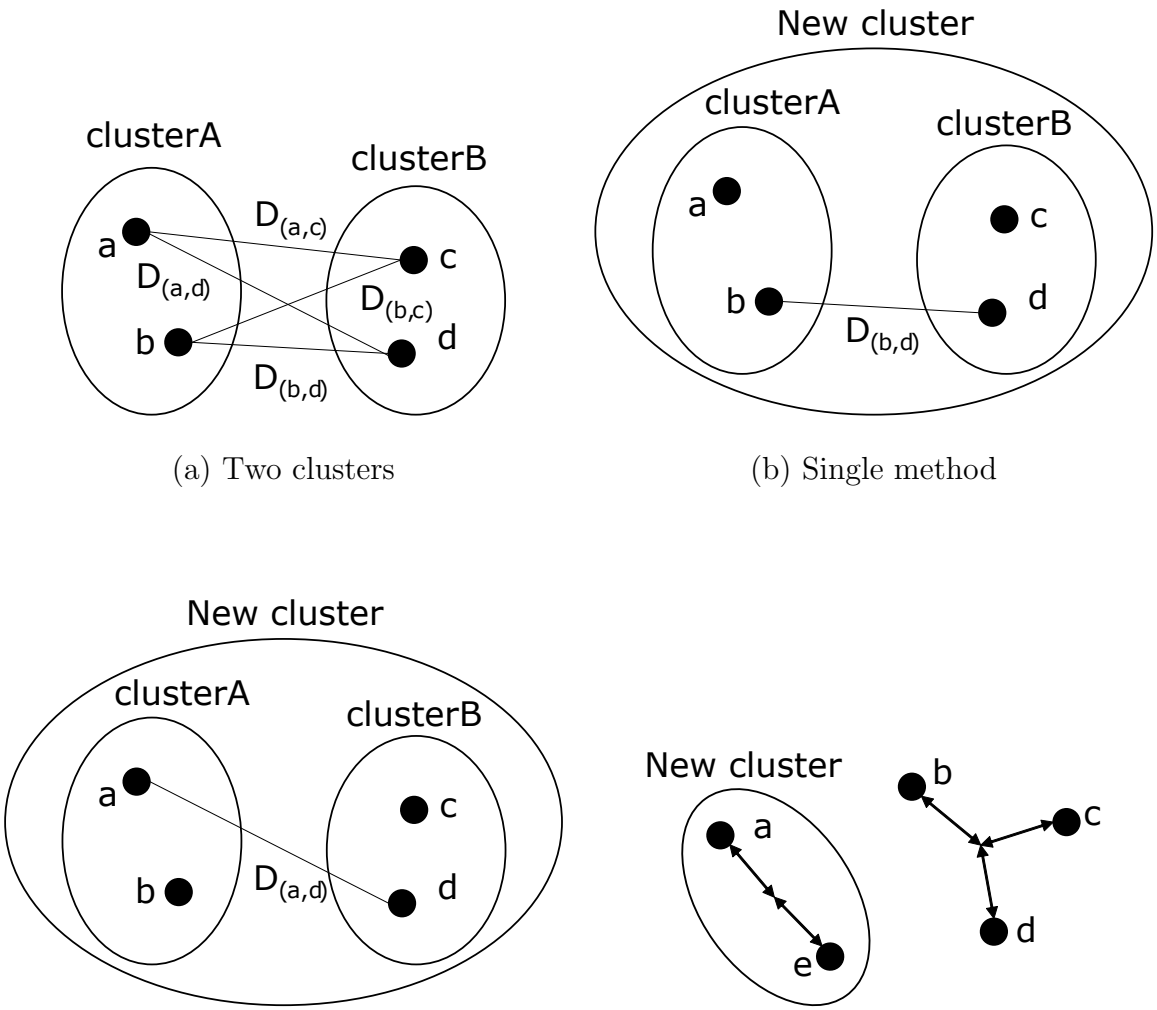
2 つのクラスターの中で、最も遠いデータ間の距離をクラスター間の距離とする方法。Figure 3.3(a) では、距離 $D_{(a,d)}$ をクラスター A とクラスター B の距離として、Figure 3.3(c) のクラスターを形成する。

群平均法

2 つのクラスターの中のすべてのデータについて距離を求め、それらの距離の平均値をクラスター間の距離とする方法。Figure 3.3(a) では、距離 $D_{(a,c)}$, $D_{(a,d)}$, $D_{(b,c)}$, $D_{(b,d)}$ の平均をクラスター A とクラスター B の距離として、新しいクラスターを形成する。

ウォード法

2 つのクラスターを融合した際に、同クラスター内の分散と他クラスター間の分散の比を最大化するようにクラスターを形成していく方法。Figure 3.3(d) の場合、データ a、e からなるクラスターを形成する。



(a) Two clusters (b) Single method

(c) Complete method (d) Ward method

Figure 3.3　Cluster

3.3　python

python は、1991 年にオランダ人のグイド・ヴァン・ロッサム氏によって開発された言語である。オブジェクト指向に対応したスクリプト言語であり、オープンソースライセンスで提供されている。文法がシンプルで読みやすく、複雑な処理を少ないコードで実現できるという特徴がある。また、豊富なライブラリが用意されており、基本的な数値計算から機械学習まで、様々な処理を行うことができる。中でも代表的なライブラリが、numpy である。numpy は数値計算用のライブラリで、特に行列計算に関して多くの関数が用意されている。このライブラリを使うことで、多次元行列の生成はもちろん、転置行列の生成や行列同士の内積の算出といった処理を高速で行うことができる。また、これらの処理は一行で完結する。近年、IT 業界を中心に人気が上昇しており、IEEE が公表した 2018 年度の人気プログラミング言語のランキングでは一位を獲得した。[4)]

3.3.1 MeCab

python で形態素解析を行う方法はいくつかある。janome というライブラリを使用した場合、インストールするだけで簡単に形態素解析ができる半面、解析の速度が遅いという問題がある。そこで本研究では、python から MeCab を呼び出し、形態素解析を行うようにした。

3.3.2 TfIdf

python では、scikit-lean というライブラリを使用すれば、簡単に TfIdf を計算することができる。しかし、標準で出力される値は、正規化されていたり、Idf に 1 が足されたりしているため、式 (3.1), (3.2) 通りの計算がされない。計算式は引数で指定することができるが、間違いを防止するため、TfIdf は自作の関数で計算した。

3.3.3 LSA

python では、scikit-lean ライブラリの TruncatedSVD という関数を使用することで、LSA によって次元削減された行列を出力できる。また、累積寄与率も同時に出力することが可能である。しかし、この関数の出力結果は、毎回変動する。そのため、numpy の svd 関数により左特異（ターム）ベクトル、特異値、右特異（文書）ベクトルを取得し、それらを基に式 (3.10) を用いて LSA を行う方法を採用した。

3.3.4 LDA

python では、scikit-lean ライブラリと gensim ライブラリの二つで LDA を行う関数が提供されている。どちらでも LDA を行うことができるが、本研究では、gensim ライブラリを使用した。このライブラリでは、辞書、コーパス、TfIdf モデルを順に作成し、それらを基に LDA モデルを作成する。作成した LDA モデルからは各単語が各トピックに属する確率を抽出することができるため、その情報から次元削減用の行列を作成した。

3.3.5 クラスター分析

python では、scipy というライブラリに含まれている linkage という関数を使用することで、クラスター分析を行うことができる。クラスター分析を行う際は、引数によって方法と距離の定義を指定する。また、fcluster という関数を使用すれば、各シラバスを指

定したクラスター数に分類することが可能である。さらに、dendrogram という関数を使用することで、クラスター分析した結果をデンドログラムとして出力できる。

3.4 日本語 WordNet

3.4.1 日本語 WordNet による単語間概念距離の算出 [5)]

日本語 WordNet とは、国立研究開発法人情報通信研究機構 (NICT) によって開発された、日本語の概念辞書のことである。英語 WordNet を基に構築されているため、日本語 WordNet で見つからない単語でも、英語版で検索すると見つかることがある。日本語 WordNet では、個々の概念が synset という単位にまとめられており、他の synset と意味的に結びついている。本研究では、シラバスに出現する各単語について、単語間の概念距離を求めるために使用した。

日本語 WordNet では、ルートの synset から下位の synset が木構造のように存在しており、概念間の関係を表している。この synset の関係から、単語間の概念距離を求める。ただし、ルートの synset は複数存在し、必ず上位下位の関係があるわけではない。そのため、概念距離を求める方法は、二つの単語の synset の関係によって次のように変化する。ここで、ある単語 a, b について、ルート synset からそれぞれの synset までの段数を L_a, L_b、二つの共通の synset の段数を C_{ab} とする。

同一の synset に存在する場合

概念距離は 0 とする。

異なる synset に存在する場合

概念距離は式 (3.15) より算出する。

異なる synset に存在し、synset 間に複数のルートがある場合

C_{ab} が最大となるようにルートをとり、C_{ab} が一致するものの中で式 (3.15) の最小値を概念距離とする。

複数の synset に属する場合

それぞれの距離を式 (3.15) より算出し、最小値を概念距離とする。

synset の関係が見つからない場合

概念距離は 1 とする。

ある二つの単語が持つ概念の類似度を $S_{a,b}$ とすると、L_a, L_b, C_{ab} を用いて次の式で表

せる。

$$S_{a,b} = \frac{2C_{ab}}{L_a + L_b} \tag{3.14}$$

式 (3.14) は最大値が 1 になるよう正規化されているため、式 (3.15) によって、概念の類似度 $S_{a,b}$ を概念距離 $D_{a,b}$ に変換できる。

$$D_{a,b} = 1 - S_{a,b} \tag{3.15}$$

3.4.2 クラスター分析による次元削減

クラスター分析は、本来次元削減を行う技術ではない。しかし、概念距離を基に単語のクラスター分析を行えば、単語を複数のクラスターに分類することができ、その結果を基に次元削減ができる。この手法をクラスター分析による次元削減とする。

単語間の概念距離をまとめた行列を対象としてクラスター分析を行うことで、各単語がどのクラスターに属しているかということが分かる。また、この情報と、式 (3.15) を用いて算出した単語間の類似度を基に計算を行うことで、式 (3.16) に示すクラスター-単語行列が生成される。ここで、式 (3.16) の $C_i(i = 1, 2, \ldots, N)$ は各クラスター、$w_j(j = 1, 2, \ldots, M)$ は各単語を表している。$a_{i,j}$ の計算式については、昨年までの研究で使用された計算式 [6] に加え、本研究では、新たに有効と思われる二種類の計算式を使用する。その三種類の式を、それぞれ式 (3.17), (3.18), (3.19) に示す。式 (3.17) はこれまでに使用された計算式であり、本研究における基準となる。また、新しく使用する式 (3.18) と式 (3.19) は、それぞれクラスターに属する名詞とそうでない名詞をより差別化すること、クラスターに属する属さないにかかわらず各名詞間の概念距離を全て利用することを目的としている。

$$CW = \left(\begin{array}{c|cccc} Term & w_1 & w_2 & \cdots & w_M \\ \hline C_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,M} \\ C_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N & a_{N,1} & a_{N,2} & \cdots & a_{N,M} \end{array}\right) \tag{3.16}$$

$$a_{i,j} = \begin{cases} \frac{1}{|C_i|} \sum_{k \in C_i} S_{k,j} & (w_j \in C_i) \\ 0 & (w_j \notin C_i) \end{cases} \tag{3.17}$$

$$a_{i,j} = \begin{cases} 1 & (w_j \in C_i) \\ 0 & (w_j \notin C_i) \end{cases} \tag{3.18}$$

$$a_{i,j} = \frac{1}{|C_i|} \sum_{k \in C_i} S_{k,j} \tag{3.19}$$

この行列に、単語-文書行列を掛け合わせることで、クラスター-文書行列を作成できる。これによって、指定したクラスター数まで次元を削減することができる。

第4章　実験

本校電気情報工学科の科目のシラバスを対象とし、次元削減をした上で、類似度計算を行った。ただし、五年生の時に両コース合同で受講する選択科目のシラバスについては、二つのうち ID が若い方のみを取り扱った。対象となるシラバスは全 122 種であり、そのうち専門科目は 73 種であった。また、類似度計算の際は、形態素解析によって得られた語のうち、名詞のみを取り扱った。対象となる名詞は、総数 3210 語であった。

4.1　実験の準備

4.1.1　環境構築

実験を始めるにあたり、まずは形態素解析や類似度計算を行うための環境を構築した。具体的には、python で使用するライブラリと MeCab のインストールを行った。ライブラリは、python に標準で付属している pip というツールを使ってインストールを行った。しかし、MeCab に関しては、インストールするだけでは python から呼び出すことができない。そのため、有志によって作成された mecab-python-windows というバインディングを導入し、python から MeCab を呼び出せるようにした。

4.1.2　シラバスのデータの取得

シラバスのデータは、Web シラバスの一覧が載っている Web ページのリンクから各シラバスのページへ飛び、それぞれのテキストデータを抽出するという方法で取得した。抽出したデータは、正規表現を利用して無駄なテキストや空白を省き、テキストファイルに保存した。ファイルに保存する際は、文字コードを python で標準となっている utf-8 にした。この操作では、Web ページのテキストを取得する requests と、各シラバスへのリンクを取得する BeautifulSoup という二つのライブラリを使用した。

4.1.3　シラバスのデータの形態素解析

取得したシラバスのデータに対して、MeCab を使用して形態素解析を行った。また、その結果を以下に示す形式で mecab ファイルに保存した。

表層形 品詞, 品詞細分類 1, 品詞細分類 2, 品詞細分類 3, 活用型, 活用形, 原形, 読み, 発音

今回の実験では、標準で用意された MeCab の辞書を使用して形態素解析を行った。そ

のため、専門用語を中心に、分割される名詞がいくつか存在した。本来であれば、これらの名詞は一つの名詞として認識されるのが理想的である。しかし、分割された名詞を全て特定するには、3000 を超える名詞を一つ一つ確認する必要がある。ただ、もしこの作業で漏れが生じてしまうと、科目ごとに条件のばらつきが出てしまう。また、専門用語を一つの名詞として扱うと、日本語 WordNet で発見できなくなるといった問題も発生する。これらの理由から、今回は標準の辞書にて形態素解析を行った。

4.1.4 TfIdf の計算

mecab ファイルに保存されたデータを読み込み、自作の関数で TfIdf を計算した。具体的な計算手順については、以下に示す通りである。また、この操作で得られる行列は、式 (4.1) に示す形式をとる。ここで、doc, w はそれぞれ N 個の文書、M 個の名詞を示し、I_{doc_i,w_j} は doc_i における w_j の TfIdf 値を表す。本研究においては、文書数 N が 122、名詞の数 M が 3210 であった。

一般的には、式 (4.1) は名詞-文書行列の形式をとるが、python においては文書-名詞行列の形式で扱うのがスタンダードとなっている。また、ライブラリもその形式を前提としているため、本研究では一般的な行列を転置した形式をとる。また、これ以降の作業についても、行列は同様の形式で扱う。

1. mecab ファイルを読み込むと同時に、名詞をすべて格納した二次元リストと、名詞の種類を格納した一次元リストを作成する。
2. collections ライブラリを用いて名詞ごとに出現回数をカウントし、Tf の二次元リストを作成する。
3. 同様のライブラリを使用し Idf の一次元リストを作成する。
4. Tf のリストと Idf のリストをかけ合わせ、TfIdf の行列を作成する。

$$\left(\begin{array}{c|cccc} Term & w_1 & w_2 & \cdots & w_M \\ \hline doc_1 & I_{doc_1,w_1} & I_{doc_1,w_2} & \cdots & I_{doc_1,w_M} \\ doc_2 & I_{doc_2,w_1} & I_{doc_2,w_2} & \cdots & I_{doc_2,w_M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ doc_N & I_{doc_N,w_1} & I_{doc_N,w_2} & \cdots & I_{doc_N,w_M} \end{array}\right) \tag{4.1}$$

4.2 データの次元削減

4.2.1 主成分数の決定

python の scikit-learn ライブラリで用意されている関数を使用し、次元削減を行った際の累積寄与率を求めた。その結果、主成分が 50 の時に累積寄与率が 80 パーセント以上になったため、主成分の数を 50 とした。本来のシラバスの数が 122 個であるため、これから行う操作によって、72 次元分の削減をすることができる。

4.2.2 LSA による次元削減

TfIdf 値が格納された文書-名詞行列に対し、LSA による次元削減を行った。この操作によって得られた行列は、コンマ区切りでテキストファイルに保存した。

4.2.3 LDA による次元削減

作成した LDA モデルから得られた各名詞が各トピックに属する確率を基に行列を作成し、その行列を文書-名詞行列に掛け合わせることで、次元削減を行った。ここで、LDA モデルを作成する際のパラメータは、すべてデフォルトの値に設定した。この操作によって得られた行列は、コンマ区切りでテキストファイルに保存した。

4.2.4 クラスター分析による次元削減

クラスター分析による次元削減を行うにあたり、まずは式 (3.15) を用いて名詞間の概念距離を計算した。また、その結果をタブ区切りでテキストファイルに保存した。この際、3210 個の名詞のうち 988 個が日本語 WordNet に登録されておらず、それらについては概念距離を計算しなかった。その後、得られた名詞間の概念距離を基に、次元削減用のクラスター-名詞行列を作成した。具体的な手順については、以下に示す通りである。

1. 式 (3.15) を用いて名詞間の類似度を計算し、行列を作成する。
2. クラスター分析によって、名詞を 50 のクラスターに分類する。
3. クラスター分析の結果にそれぞれ式 (3.17), (3.18), (3.19) を適用し、式 (3.16) に示す行列を作成する。

名詞のクラスター数を 50 に設定したのは、LSA や LDA の時と次元削減後の次元を合わせるためである。また、クラスター分析については最遠隣法とウォード法の二つの手法で行った。その理由は、概念距離はユークリッド距離やマンハッタン距離とは違い、

最適な手法が判断できないからである。そこで両者を比較し、どちらがより有効といえるか検討した。最終的には、TfIdf 値が格納された文書-名詞行列に次元削減用のクラスター-名詞行列を転置して掛け合わせ、次元削減を行った。本研究では、クラスター分析の手法が二種類、クラスター-名詞行列生成の式が三種類であるため、六種類の行列が生成された。

4.3 類似度の計算

LSA, LDA, クラスター分析のそれぞれの手法によって次元削減された行列について、cos 類似度を計算した。また、その結果を行列として出力し、コンマ区切りにしてテキストファイルに保存した。

4.4 実験結果

4.4.1 実験結果の評価方法

実験結果については、シラバス同士の cos 類似度が格納された行列を対象としてウォード法を用いたクラスター分析を行い、その結果出力されるデンドログラムを比較することで評価する予定であった。しかし、今回は類似度計算の対象としたシラバスの数が多く、そのままではデンドログラムによる可視化が困難である。そのため、特に専門科目のシラバスについてクラスター分析を行い、出力されるデンドログラムで評価を行うこととする。また、クラスター分析による次元削減を行ったデータについても種類が多いため、式 (3.18), (3.19) を適用した場合については、代表してウォード法の結果を取り扱うこととする。評価の際は、LSA による次元削減を行ったデータのデンドログラムを基準とする。

4.4.2 LSA による次元削減を行った際の科目間類似度

LSA による次元削減をしたデータについて類似度計算を行い、デンドログラムとして出力した結果を Figure 4.1 に示す。

まず、科目名の最後に付くアルファベットやローマ数字を除き同じ名前をもつ科目に着目すると、ほぼ全ての科目が最初に結合している。また、実験系やプログラム系、物理学系、回路系の科目についても早いタイミングで結合し、クラスターを形成している。一番大きい二つのクラスターで比較すると、大まかにプログラミングや実験といった実

践形式の科目と、講義形式の科目に分類されている。ただし細かく見ていくと、画像処理とプラズマ工学、発変電工学のように、他分野の科目との結合が早い科目も数点存在する。

総合的に判断すると、ほぼ同系統の科目同士でクラスターを構成し、大きな分類も科目の形式に沿っているため、基準として問題ないといえる。

4.4.3 LDA による次元削減を行った際の科目間類似度

LDA による次元削減をしたデータについて類似度計算を行い、デンドログラムとして出力した結果を Figure 4.2 に示す。

LSA と比較すると、クラスターの構成が大きく異なっている。LDA では、全体的に結合するタイミングが早く、早い段階で大きな二つの分類がされている。しかし、大きな二つの分類については、LSA のような明確な基準が見受けられない。実験系やプログラム系の科目は、変化が少ないところもあるが、回路系や物理学系の科目は属するクラスターがかなり変化している。さらに、情報コースと電気コースの科目が混在しているクラスターも増えている。ただ、同名の科目については、結合が早いものが多くなっている。

総合的に判断すると、結合が早いという特徴があるが、より適切な分類がされているとは言い難い。

4.4.4 クラスター分析による次元削減を行った際の科目間類似度

クラスター分析を使用して次元削減したデータについて類似度計算を行い、デンドログラムとして出力した結果を Figure 4.3 から Figure 4.6 に示す。

まずは Figure 4.3 に示される、ウォード法と式 (3.17) を用いて次元削減を行ったデータに着目する。LSA と比較すると、同名、同系統の科目がかなり分離してしまっている。特に電気回路や物理学系の科目が顕著であり、同クラスターに属している科目が少なくなっている。また、LSA と共通するものも多いが、他分野との結合が早い科目も増えている。ただ、これまでの手法と大きく異なっている点もあり、それが大きなクラスターの構成である。この手法においては大きなクラスターが三つあり、それぞれに属する科目が大まかに情報系、電気系、実験系に分かれている。LSA や LDA では大きなクラスターでとらえてもこのような分かれ方をしておらず、情報系と電気系が交互に現れる箇

所も存在していた。そのため、一概に不適切な分類がされているとは言い難い。

次に Figure 4.4 に示される、ウォード法と式 (3.18) を用いて次元削減を行ったデータに着目する。傾向としては式 (3.17) を用いた時と似た部分もあるが、細かいクラスターの構成がよりばらけてしまっている。これまで比較的早い段階で結合していたプログラミングや実験系の科目が、一部別のクラスターに属しているという結果になった。既に分離していた電気回路や物理学系の科目もさらにばらけ、別々のクラスターに属している。大きなクラスターについては、電気系や情報系がある程度固まっているが、実験については明確な分類がされなくなっている。

さらに Figure 4.5 に示される、ウォード法と式 (3.19) を用いて次元削減を行ったデータに着目する。この手法は他の手法と比べ、クラスターの内容以前にシラバスの結合スピードが大きく異なっている。これまではクラスター間の距離が約 8 となる部分で全シラバスが結合していたのに対し、この手法ではクラスター間の距離が約 0.35 となる部分で全シラバスが結合する。こうなるのは、シラバス同士の類似度がほぼ 99 パーセントを超えているからである。式 (3.19) を用いて次元削減用の行列を作成した場合、行列の値が似通ってしまい、このような結果が現れる。肝心なシラバスの分類結果については、同系統の科目の分離がさらに激しくなっている。特に変化が現れたのは実験系の科目である。これまでは一部を除き同一クラスターに属していたが、今回は点在するようになった。その他にも、同名科目が最初に結合しなくなったり、同系統科目で構成されたクラスターが大幅に減少したりしている。ただ、大きなクラスターで見たときに、情報系と電気系で分かれるという特徴は多少残っている。

最後に Figure 4.6 に示される、最遠隣法と式 (3.17) を用いて次元削減を行ったデータに着目する。LSA と比較すると、細かいクラスターについて似た部分が数か所あるのみで、全体的には大きく構造が変わっている。例えば、電子物性と応用物理Ⅱ、エネルギー変換工学とパワーエレクトロニクスといった部分は結合する早さが似ているが、その他については類似する点が少ない。また、同系統の科目、特に回路系やプログラム系が同じクラスターに属していないことが多い。クラスター間の距離の測定方法以外の条件が同じ Figure 4.3 と比べても、類似する点が少ない。細かいクラスターはともかく、大きなクラスターで見ても、情報系と電気系にある程度分かれているといった特徴もない。そのため、最遠隣法を用いた分類は適当でないといえる。

総合的に判断すると、どの手法においても LSA ほどの細かいクラスターの分類ができ

ず、より適切な手法が見つかったわけではないといえる。ただし、ウォード法と式 (3.17) を用いた手法であれば、科目を大きく分類することができ、コースごとの類似度を把握するのには向いているといえる。

4.5 考察

4.5.1 LSA, LDA, クラスター分析の比較

今回の実験では、基準としていた LSA による次元削減が最適という結果になった。しかし、本来であれば、LSA を発展させた手法である LDA による次元削減の方が、より良い結果をもたらすはずである。そうならなかったのは、今回のパラメータの設定が次元削減に向いていなかったからだと考えられる。LDA モデルを作成する際には、反復する文書数や、取り扱う確率の最低値等を設定することができる。それらを変化させることで、より良い次元削減となる可能性がある。また、各トピックにおける各名詞が属する確率を全て取り扱ったことも、次元削減が不適切になった原因だと考えられる。LDA モデルから各名詞が各トピックに属する確率を抽出しようとすると、デフォルトでは確率の高い上位 10 個のみが抽出される。しかしこの場合だと、次元削減後にすべての要素が 0 になってしまう文書が現れてしまったため、各トピックに属する確率が低い名詞も、計算に使用せざるを得なかった。もし何らかの方法で確率の低い名詞を無視できれば、不適切ではなくなると考えれらる。また、クラスター分析による次元削減については、どの方法も LSA ほど適切な次元削減ができていなかった。これは、手法自体が有効でないという可能性もあるが、日本語 WordNet に登録された名詞が少なかったことも原因であると考えられる。今回は、シラバスに出現した名詞のうち、約 30 パーセントの名詞が日本語 WordNet に登録されておらず、概念距離を求めることができなかった。もしこれらの名詞について、英語に置き換えた上で英語 WordNet を用いて概念距離の計算ができれば、クラスター分析の結果も変化する。そうなれば、次元削減がより良くなる可能性がある。しかし、日本語 WordNet に存在しない名詞をすべて英語に変換する場合の作業量は多く、英語でも見つからない可能性があるという欠点がある。

4.5.2 クラスター分析同士の比較

クラスター分析による次元削減においては、ウォード法と式 (3.17) を用いた次元削減が最適であった。これより、シラバスのデータの次元削減においても、最遠隣法と比べ

ウォード法が有効であることが確認できた。また、クラスター-名詞行列を作成する際に使用する式を、式 (3.18), (3.19) にすると改悪となってしまうことが発覚した。そこで、それぞれの問題点について考察する。まず式 (3.18) であるが、この式の問題点は、クラスター内での重要度が考慮されていないことである。あるクラスターに属する名詞は、ほとんどの場合でそれぞれの重要度が異なるが、この式では等しく 1 になる。そのため、重要度が低い名詞も重要度が引き上げられてしまい、文書の各クラスターへの依存度が大きくなってしまう。本来はクラスターに属する名詞とそうでない名詞をより差別化することを目的としていたが、得られる恩恵より悪影響の方が大きくなってしまったと考えられる。また式 (3.19) であるが、この式の問題点は、各クラスターに属さない名詞についても重要度を計算していることである。これにより、各名詞の重要度が平均化され、全ての文書の各クラスターへの依存度が似通ってしまうという結果になった。この式にも、各名詞間の概念距離を全て活用し損失を抑えるという目的があったが、悪影響の方が大きくなってしまったと考えられる。

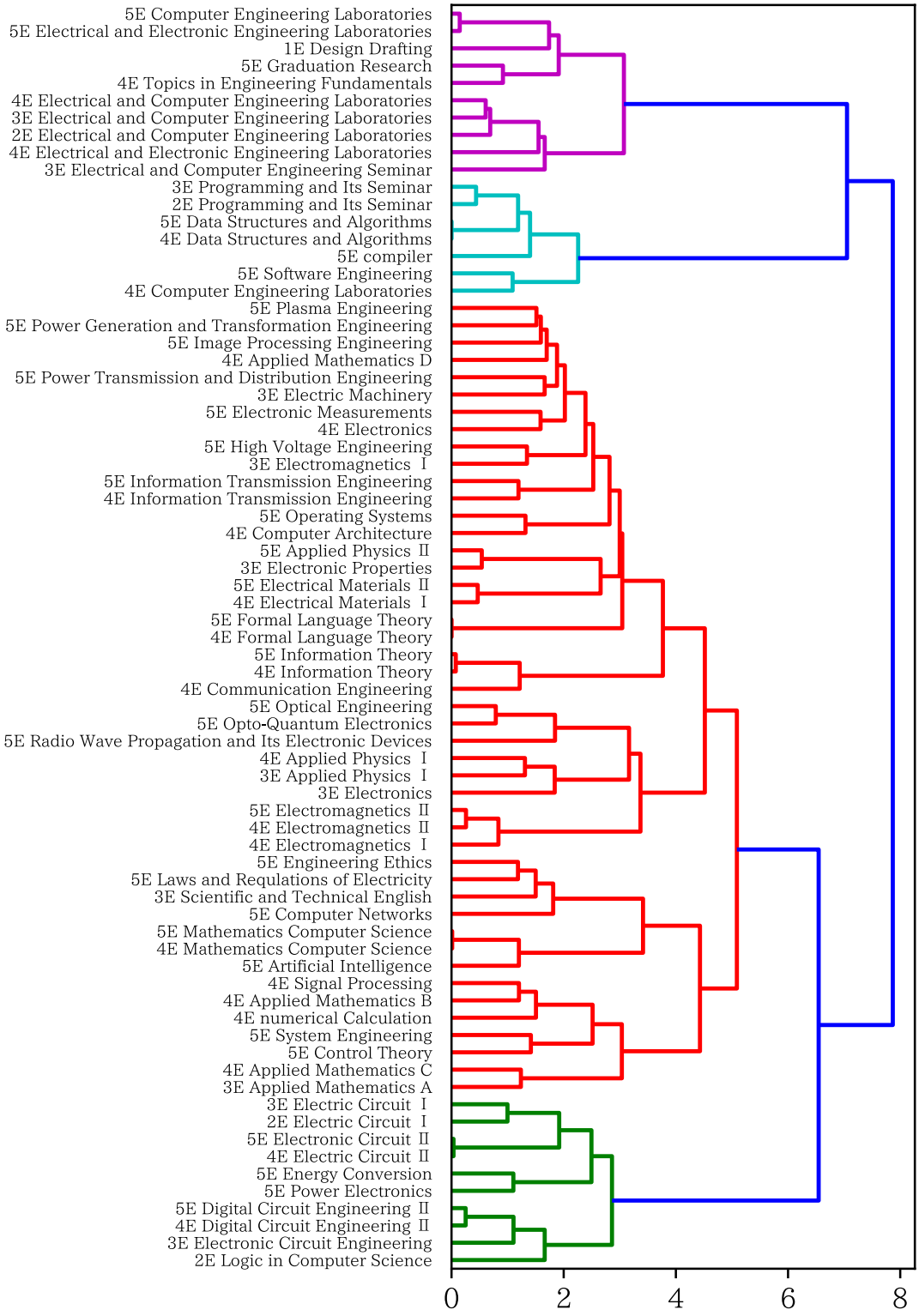


Figure 4.1 Result of dimension reduction by LSA

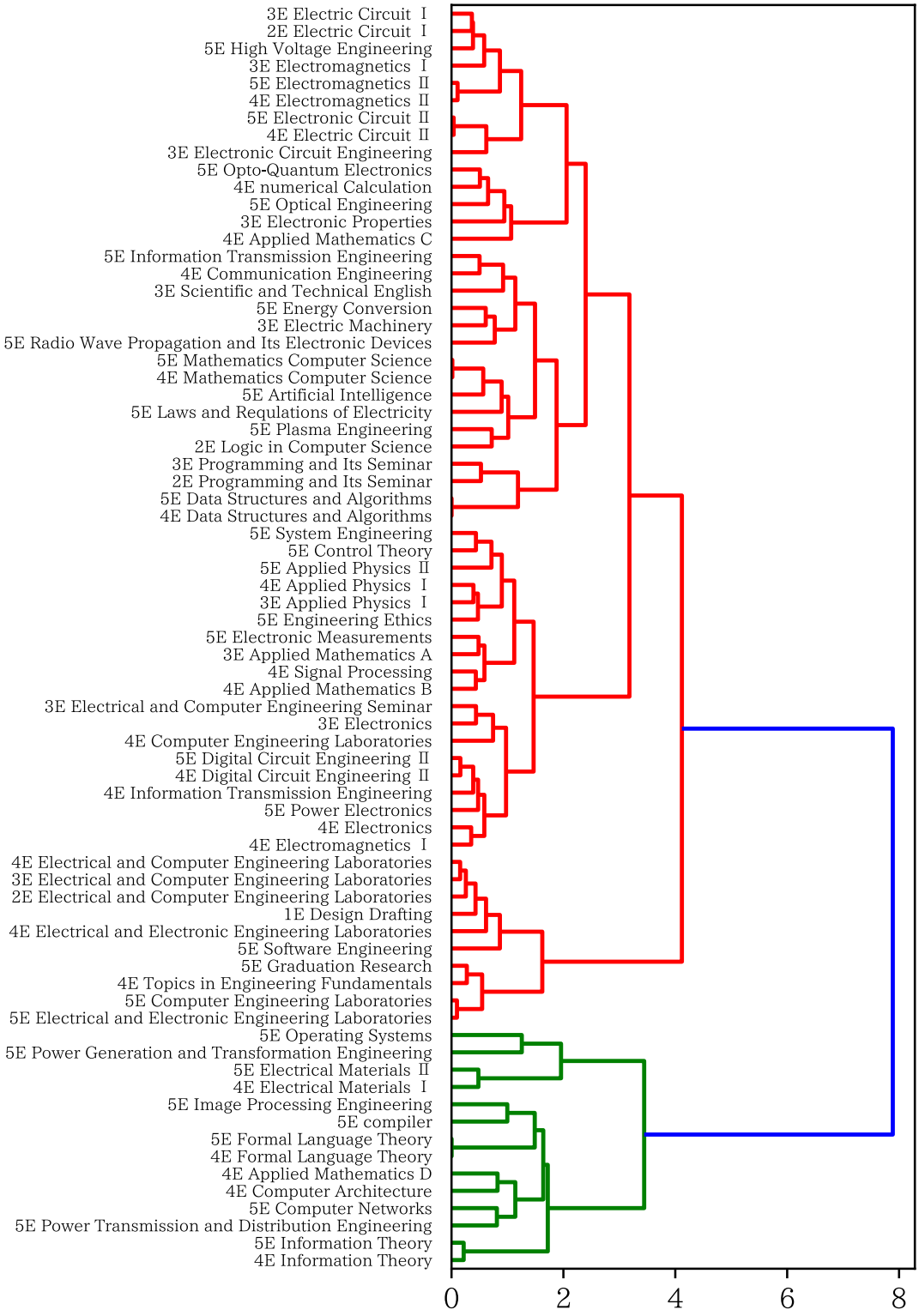


Figure 4.2 Result of dimension reduction by LDA

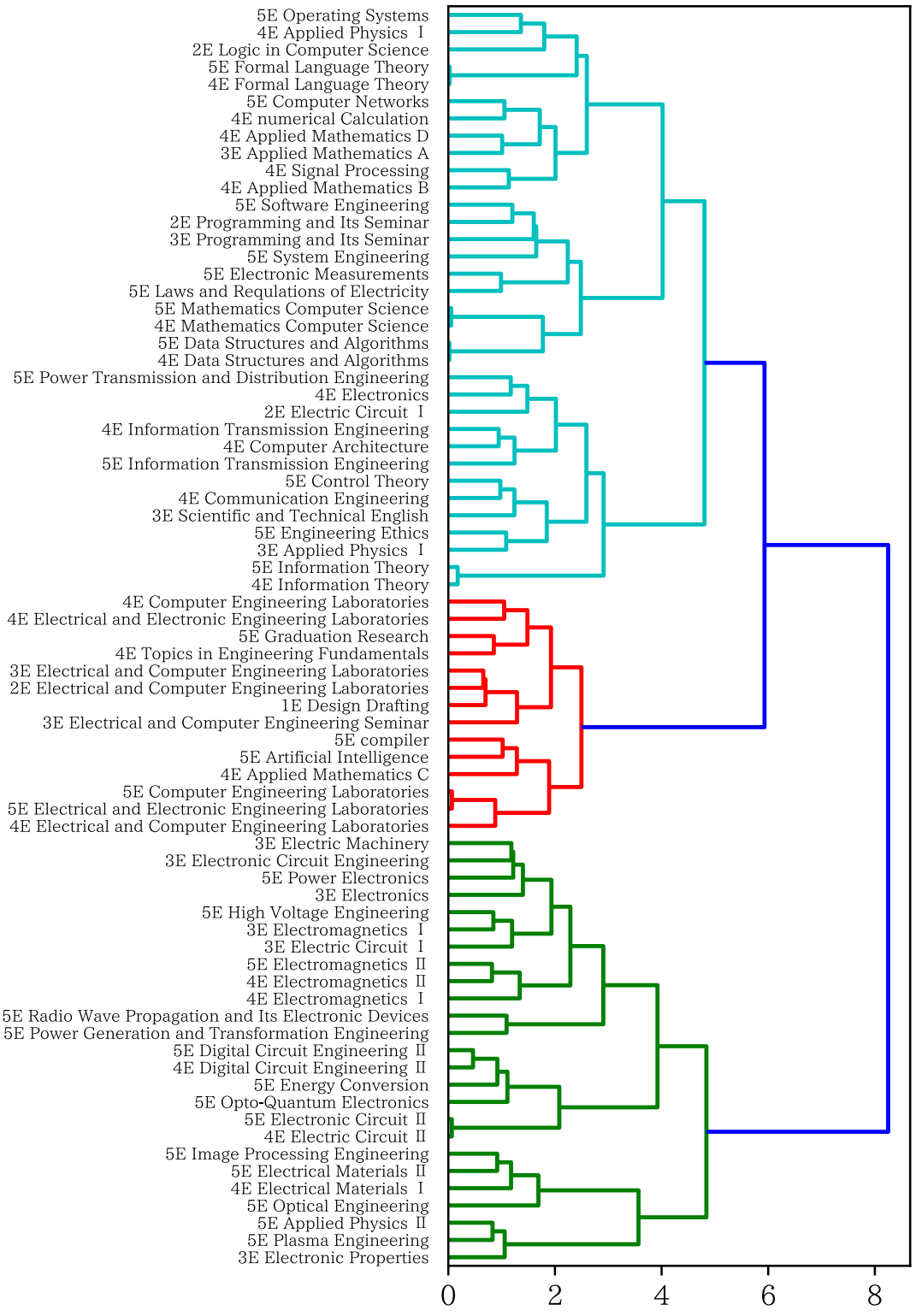


Figure 4.3 Result of dimension reduction by cluster analysis using Ward method and Eq.(3.17)

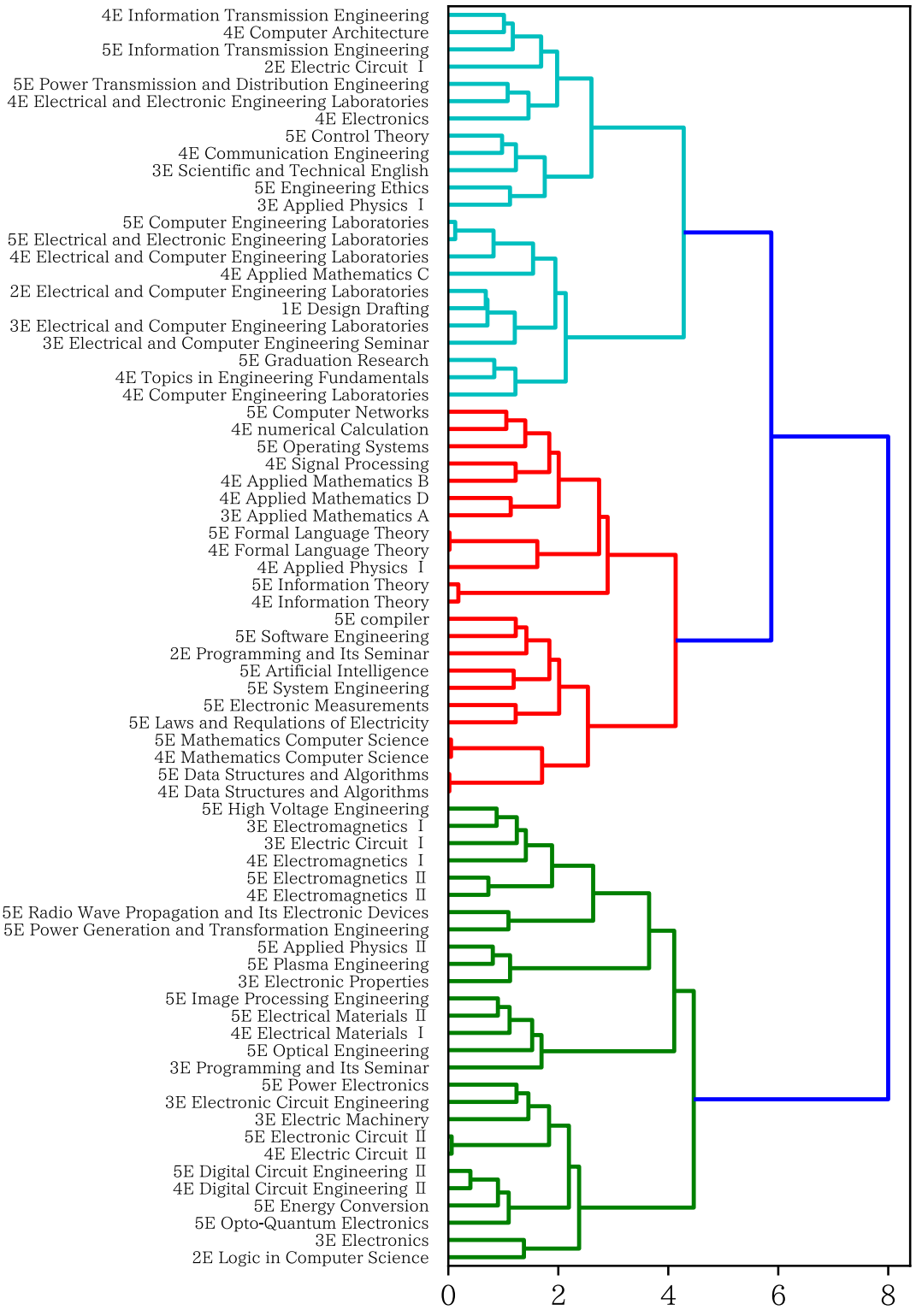


Figure 4.4 Result of dimension reduction by cluster analysis using Ward method and Eq.(3.18)

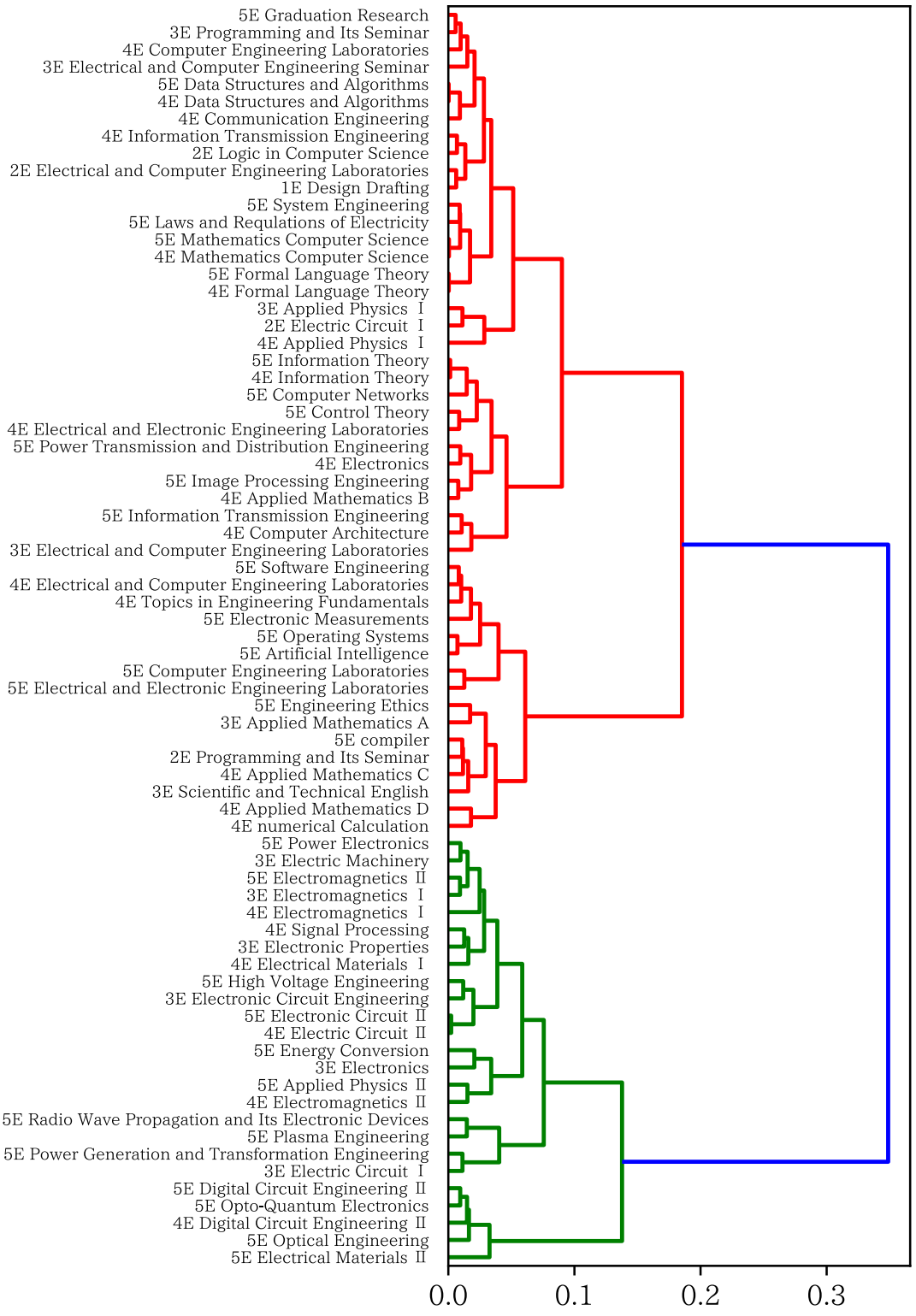


Figure 4.5 Result of dimension reduction by cluster analysis using Ward method and Eq.(3.19)

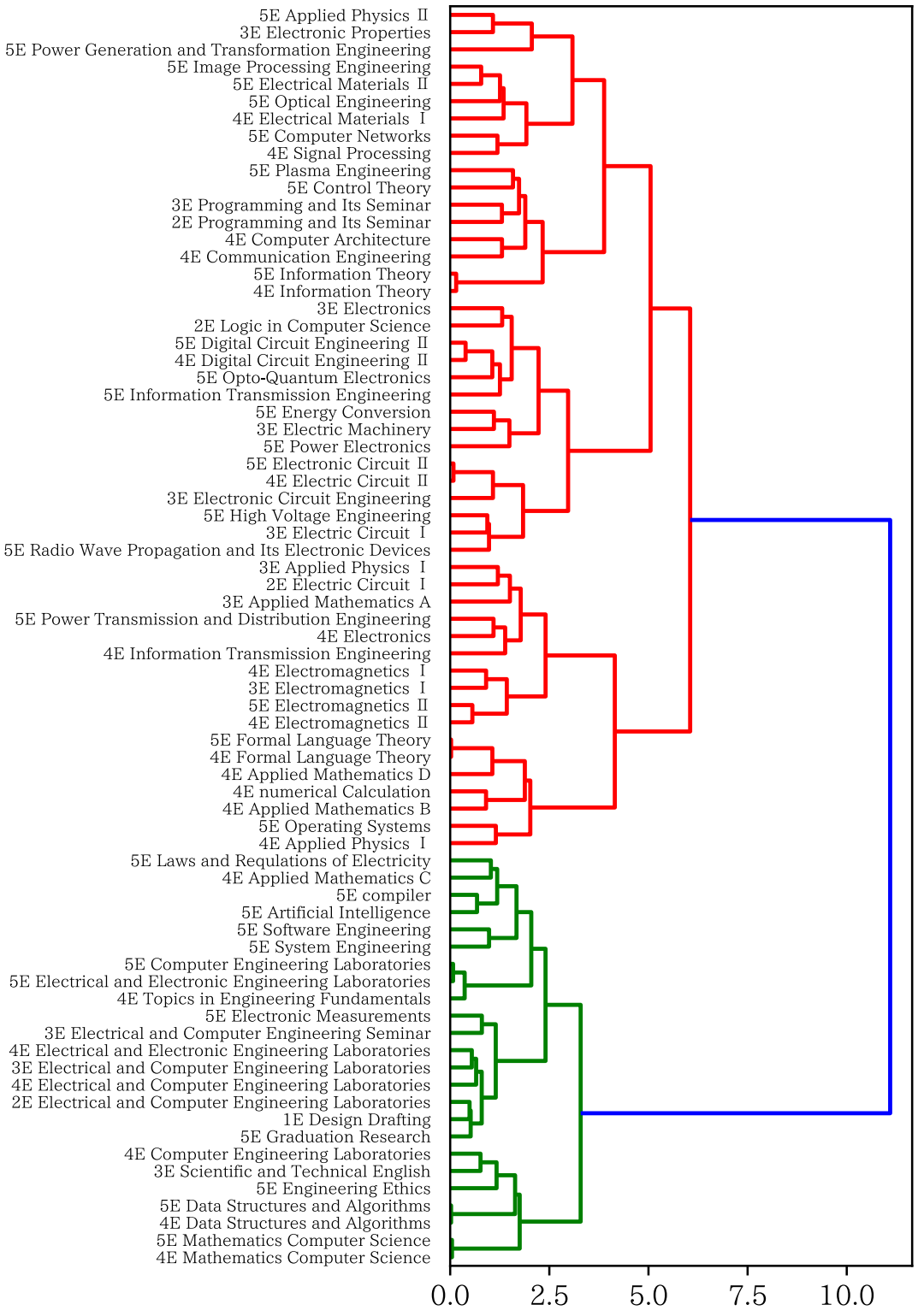


Figure 4.6 Result of dimension reduction by cluster analysis using complete method and Eq.(3.17)

第5章　結論

本研究では、本校電気情報工学科の科目のシラバスについて、データの次元を削減した上で、類似度計算を行った。また、この作業で用いたデータの次元を削減する各手法について、それぞれの有効性を検討した。

手順として、まずはシラバスのデータの取得、形態素解析、TfIdf 値の算出、日本語 WordNet を用いた名詞間の概念距離の計算を行った。その後、得られたデータに対して LSA, LDA, クラスター分析による次元削減を行った。ここで、クラスター分析による次元削減は、クラスター間の距離の測定方法や次元削減用行列を作成する式を変更し、複数の手段にて行った。最終的には、次元削減後のデータについて cos 類似度を算出し、専門科目の類似度をデンドログラムによって可視化した。

これらの作業により得られたデンドログラムを比較したところ、LSA による次元削減が最適という結果になった。予想では、LSA を発展させた LDA による次元削減がより良い結果になると考えていたが、実際はそうならなかった。こうなった原因については、LDA モデルを作成する際のパラメータや、各単語の各トピックに属する確率の最低値の設定が適切でなかったからだと考えらえる。また、クラスター分析による次元削減も、LSA と比較してより良くなったわけではなかった。こちらの原因については、日本語 WordNet に登録されていない名詞が多く、それらをクラスター分析に使用できなかったからだと考えられる。しかし、ウォード法を用いたときに関しては、大きなクラスターの中身が大まかに分野ごとに分類されていた。そのため、部分的に見れば、LSA の問題点を解消しているともいえる。

また、クラスター分析を用いた次元削減手法同士の比較では、ウォード法と式 (3.17) を用いた手法が最適な結果をもたらした。これより、シラバスのデータの次元削減についても、一般的に分類感度が高といわれるウォード法が有効であるといえる。また、次元削減用の行列を作成する式については、式 (3.18), (3.19) は改悪となり、恩恵が得られないといえる。

総合的に今回の実験を振り返ると、有効といえる次元削減手法を新しく発見することはできなかった。しかし、それぞれの手法に関して LSA とは違う特徴や利点があるため、改善をしていくことで有効な手法になる可能性がある。特にクラスター分析については、三つの方法で改善できる。一つ目は概念距離を算出できる名詞を増やすことである。名

詞を英語に置き換え、英語 WordNet を用いてより多くの名詞の概念距離が算出できれば、より適切な次元削減になると考えられる。二つ目はクラスター数の変更である。概念距離を算出できる名詞の割合に応じてクラスター数を変化させれば、より適した分類ができると考えられる。三つ目は次元削減用の行列を作成する式の変更である。今回の実験では二つの式は改悪となってしまったが、式の改良の余地はまだあるため、さらに別の式を使えばより良い次元削減となる可能性がある。

謝辞

本研究を進めるにあたり、ご多忙中にもかかわらず適切な指導をしていただきました出口利憲先生に深く感謝申し上げます。同時に、助言や協力を頂いた同研究室並びに同クラスの皆様にも厚く御礼申し上げます。

参考文献

1) 加納学, 主成分分析, 京都大学大学院工学研究科化学工学専攻プロセスシステム工学研究室, 1997.
http://manabukano.brilliant-future.net/document/text-PCA.pdf

2) Developers.IO, 機械学習_潜在意味解析_理論編
https://dev.classmethod.jp/machine-learning/2017ad_20171220_lsa/ (2018 年 11 月 8 日アクセス).

3) 奥村学 監修, 佐藤一誠 著, 自然言語処理シリーズ 8 トピックモデルによる統計的潜在意味解析, コロナ社, 2015.

4) IEEE, The 2018 Top Programming Languages
https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages (2019 年 1 月 28 日アクセス).

5) Francis Bond, Timothy Baldwin, Richard Fothergill and Kiyotaka Uchimoto, Japanese SemCor: A Sense-tagged Corpus of Japanese, The 6th International Conference of the Global WordNet Association (GWC-2012), 2012.
http://compling.hss.ntu.edu.sg/wnja/pubs/2012-gwc-jsemcor.pdf

6) 服部修平, テキストマイニングによる文書の類似度計算に関する研究, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2017.
http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/hattori/