修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院　　情報理工学研究科　情報・ネットワーク工学専攻　博士前期課程		
氏　　名	藤原　勇二	学籍番号	1631129
論 文 題 目	多観点類似度を用いたクラスタリングに関する研究		

要　　旨

データ集合を教師データを用いた事前学習をおこなうことなくクラスタと呼ばれる部分集合に分割する手法をクラスタリングと呼ぶ。クラスタリングの基本は、類似したデータ同士を同じクラスタに所属させることである。このため、データ間の類似度の設定はクラスタリングにおいて非常に重要である。一般的に用いられる代表的な類似度としては、ユークリッド空間上の多次元ベクトルに対するユークリッド距離や cosine 類似度が知られている。Cosine 類似度は、文書データのような高次で疎なデータに対する類似度指標としてよく用いられる。

Nguyen らは cosine 類似度における原点を複数用いた多観点類似度(Multiviewpoint-Based Similarity:MVS) を提案した。そして、MVS を非階層クラスタリングに適用することで、文書データのクラスタリングにおいて優れた結果を示した。ただし、非階層クラスタリングは事前に分割するクラスタの数を人為的に指定する必要がある。

本研究では、この多観点類似度に関する 2 つのテーマを取り扱う。

1 つ目は、Nguyen らの提案した多観点な cosine 類似度を階層クラスタリングについて適用した手法の開発である。階層クラスタリングは非階層クラスタリングのように事前に分割するクラスタ数を指定する必要がなく、階層的な分割構造を抽出できる。ただし MVS は cosine 類似度より計算量が大きいため、階層クラスタリング全体の計算量を悪化させる恐れがある。そこで提案手法では、クラスタ間類似度の計算を高速化する手法を開発し、一般的な階層クラスタリングと同様の計算量 O(mn2+n2logn)でのクラスタリングを実現した。さらに文書データを用いた実験により、MVS を用いた階層クラスタリングが既存手法と同程度の計算時間で、より高い分類精度を示すことを確認した。

2 つ目は、cosine 類似度以外への多観点類似度の適用である。本研究では、ユークリッド距離に対して基準点が影響を与えるような新しい距離定義である多観点距離(Multiviewpoint-Based Distance:MVD) を提案する。さらに、この MVD を、非階層クラスタリングの代表的手法である k-means に対して適用したクラスタリング手法を開発した。また、開発した MVD を用いた分割クラスタリング手法が、k-means のクラスタリング結果を改善することを実験的に示した。

平成29年度　修士論文

多観点類似度を用いたクラスタリングに関する研究

電気通信大学大学院 情報理工学研究科
情報・ネットワーク工学専攻

1631129　藤原 勇二

指導教員
古賀 久志 准教授
南 泰浩 教授

平成30年1月29日

目 次

第1章

序論

データ集合を教師データを用いた事前学習をおこなうことなくクラスタと呼ばれる部分集合に分割する手法をクラスタリングと呼ぶ．クラスタリングの基本は，類似したデータ同士を同じクラスタに所属させることである．このため，データ間の類似度をどのように設定するかは，クラスタリングにおいて非常に重要である．一般的に用いられる代表的な類似度としては，ユークリッド空間上の多次元ベクトルに対するユークリッド距離や cosine 類似度が知られている．Cosine 類似度は，文書データのような高次で疎なデータに対する類似度指標としてよく用いられる [1].

Nguyen らは cosine 類似度における原点を複数用いた多観点類似度 (Multiviewpoint-Based Similarity:MVS) を提案した．そして，MVS を非階層クラスタリングに適用することで，文書データのクラスタリングにおいて優れた結果を示した [2]．ただし，非階層クラスタリングは事前に分割するクラスタの数を人為的に指定する必要がある．

本研究では，この多観点類似度に関する 2 つのテーマを取り扱う．

1 つ目は，Nguyen らの提案した多観点な cosine 類似度を階層クラスタリングについて適用した手法の開発である．階層クラスタリングは非階層クラスタリングのように事前に分割するクラスタ数を指定する必要がなく，階層的な分割構造を抽出できる利点を有する．ただし MVS は cosine 類似度より計算量が大きいため，階層クラスタリング全体の計算量を悪化させる恐れがある．そこで提案手法では，クラスタ間の類似度を高速に更新する式を用いて一般的な階層クラスタリングと同様の計算量 $O(mn^2 + n^2 \log n)$ でのクラスタリングを実現する．また，文書データを用いた評価実験により，提案手法が MVS を使用しない通常の階層クラスタリ

ングと比較して同等の計算時間で分類精度を向上させることを示し，実データへの階層クラスタリングに対して有用であることを示す．

2つ目は，cosine 類似度以外への多観点類似度の適用である．一般に広く用いられる類似度指標として，ユークリッド距離がある．そこで，このユークリッド距離に対して基準点が影響を与えるような新しい距離定義である多観点距離 (Multiviewpoint-Based Distance:MVD) を提案する．さらに，この MVD を，非階層クラスタリングの代表的手法である k-means[3] に対して適用したクラスタリング手法を開発する．また，開発した MVD を用いた分割クラスタリング手法が，k-means のクラスタリング結果を改善することを実験的に示す．

本論文における構成を以下に示す．2 章では本論文における提案手法を論じる上で必要となる関連研究についての説明をおこなう．3 章では MVS を凝集型階層クラスタリングへ導入した提案手法について述べる．4 章ではユークリッド距離を基盤とした多観点距離の定義及び非階層クラスタリング手法へ導入した提案手法について述べる．5 章では結論及び今後の展望について論じる．

第2章

関連研究

2.1 文書データの特徴ベクトル表現

本研究では，文書データを用いてクラスタリング手法を評価する．そのため本節では，データ解析手法における文書データの表現方法について論じる．

文書データにおける特徴は，その中で出現する単語の種類やその出現頻度によって表される．そのような基準に基づく文書データの特徴表現に TF-IDF(Term Frequency-Inverse Document Frequency) がある [4]．データセットの辞書にある単語 t_i のデータセット S がもつ文書 d_j における重みについて，TF-IDF は式 (2.1) のように単語頻度 (Term Frequency:TF) と逆文書頻度 (Inverse Document Frequency:IDF) から得られる．

$$\text{tf-idf}(t_i, d_j) = \text{tf}(t_i, d_j) \times \text{idf}(t_i) \tag{2.1}$$

TF は単語 t_i の文書 d_j 中における出現度であり，式 (2.2) のように表される．

$$\text{tf}(t_i, d_j) = \frac{\text{文書 } d_j \text{における単語 } t_i \text{の出現回数}}{\text{文書 } d_j \text{における全単語の出現回数}} \tag{2.2}$$

また，IDF はデータセットにおける単語の重要度で，式 (2.3) のように多くの文書で出現するような単語について値を小さくする働きをもつ．

$$\text{idf}(t_i) = \log \frac{\text{データセットの文書数}}{\text{単語 } t_i \text{を持つ文書数}} \tag{2.3}$$

TF-IDF を用いて文書 d_j の特徴ベクトルを得る場合には，式 (2.4) のようにデータセット中で出現する m 種類の全単語 $t_i (i = 1, \ldots, m)$ について，文書 d_j に対する $\text{tf-idf}(t_i, d_j)$ を要素として持つ m 次元ベクトルを用いる．

$$\text{特徴ベクトル } d_j = \{\text{tf-idf}(t_i, d_j) \mid i = 1, \ldots, m\} \tag{2.4}$$

2.2　クラスタリング

本節では，本研究において取り扱うデータ解析手法であるクラスタリングについて述べる．クラスタリングとは，与えられたデータ集合をクラスタと呼ばれる部分集合に分割する教師なし機械学習の手法である．クラスタとは，内的結合及び外的分離の性質に基づいたデータ集合の部分集合である．ゆえにクラスタはデータの特徴量から類似性の高いものをまとめた集合であり，事前にその意味や分類基準が定義されたクラスとは異なるものである．

クラスタリングの手法は，その手続きの流れにより大きく階層クラスタリングと非階層クラスタリングの 2 種類に分けられる．

2.2.1　階層クラスタリング

階層クラスタリングとは，データ数 n 個のデータ集合について事前に分割するクラスタ数を決めることなく，全てのデータが異なる n 個のクラスタに属する状態から，同一クラスタに属する状態までの分割結果を階層的に得ることができるクラスタリング手法である．そのため，階層クラスタリングではあらかじめ分割するクラスタ数を指定する必要がなく，得られた階層的なクラスタリング結果から任意のクラスタ数での分割結果を得ることができる．

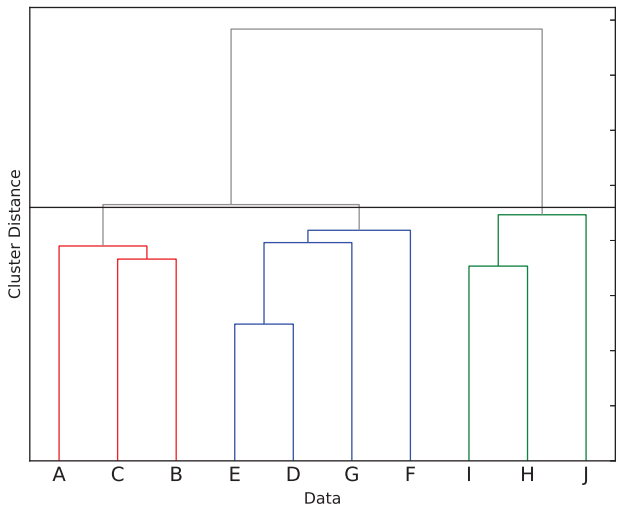

図 2–1: 階層クラスタリング結果のデンドログラム

階層クラスタリングの結果は図 2–1 のような樹形図 (デンドログラム) によって表現できる．デンドログラムにおいて縦軸はクラスタ間の距離 (非類似度) であり，この値が小さい順にクラスタがマージしている様子がわかる．任意のクラスタ数 K での分割結果は，この樹形図を K 個の部分木に分けられる高さで切断することで得られる．図 2–1 では，$K = 3$ としたときの分割結果を示している．階層クラスタリングによる分類結果の特徴としてクラスタ数 K のとき同じクラスタに所属するデータ対が，K よりも少ないクラスタ数 K' においても同じクラスタに所属することが保証されることが挙げられる．これは K' で分けた時の 1 つのクラスタがより大きな K で分けたときのクラスタ 1 個以上の和集合で構成されるためである．

階層クラスタリングは，大きく分割型階層クラスタリングと凝集型階層クラスタリングの 2 種類に分けられる．本研究では凝集型のみを取り扱う．そのため以降では階層クラスタリングはこの凝集型を指す．また，凝集型階層クラスタリングについては 2.5 節にて論じる．

2.2.2　非階層クラスタリング

非階層クラスタリングはあらかじめ定められたクラスタ数 K に対して最適なクラスタ分割をおこなう手法である．この手法ではクラスタ分割の良さを評価する目的関数の最適化問題に基づいて，この関数を最適化するようなクラスタ分割を探索する．そのことから非階層クラスタリングは分割最適化クラスタリングとも呼ばれる．ただし，n 個のデータを K 個のクラスタに分割するパターン (解の候補) は莫大な数になるため，目的関数の大域最適解を計算することは非常に困難である．ゆえに非階層クラスタリングでは，目的関数の局所最適解を発見し，その局所最適解に基づくクラスタ分割を得る．計算量が階層クラスタリングよりも高速である手法が多く存在するため，階層的な分類構造を必要としない場合にはこの非階層クラスタリングがよく用いられる．

2.3　多観点類似度

本節では，Nguyen らが提唱した cosine 類似度に関する MVS について説明する．これは高次元の単位球面上に存在する単位ベクトルを対象とする類似度である．単位ベクトルであるデータ d_i, d_j 間の cosine 類似度 $\mathrm{CS}(d_i, d_j)$ は，式 (2.5) のように

ベクトル内積として定義される．

$$\mathrm{CS}(d_i, d_j) = d_i^{\mathrm{T}} d_j \tag{2.5}$$

この式は，式 (2.6) のように，各データと原点 0 の差の内積としても表せる．これは cosine 類似度が原点 0 のみを唯一の基準点として類似性を評価していることを表している．

$$\mathrm{CS}(d_i, d_j) = (d_i - 0)^{\mathrm{T}}(d_j - 0) \tag{2.6}$$

MVS の狙いは，この基準点をデータセット中の複数の様々な点に移動させることで，cosine 類似度よりもデータ分布に適応した類似性評価を実現することである．よって，MVS は式 (2.7) のように同一クラスタ内の 2 データ d_i, d_j の類似度を n 個の全データの集合 S から d_i, d_j が所属するクラスタ r の集合 S_r を除いたものを基準点 d_h として，各 d_h とのベクトル差の内積の平均によって定義される．ここで，式中の n_r はクラスタ r に所属するデータの数である．

$$\begin{aligned}
&\mathrm{MVS}(d_i, d_j \mid d_i, d_j \in S_r) \\
&\quad = \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} (d_i - d_h)^{\mathrm{T}}(d_i - d_h) \qquad (2.7) \\
&\quad = \frac{1}{n - n_r} \sum_{d_h \in S \setminus S_r} \Big\{ \mathrm{CS}(d_i - d_h, d_j - d_h) \\
&\qquad\qquad \| d_i - d_h \| \| d_j - d_h \| \Big\} \qquad (2.8)
\end{aligned}$$

MVS は式 (2.8) のように原点の異なる cosine 類似度の重み付き平均としても表すことができる．ここで重み $\| d_i - d_h \| \| d_j - d_h \|$ は，d_i, d_j と d_h とのユークリッド距離の積である．異なるクラスタの集合に含まれる d_h に対して d_i, d_j が大きい非類似度を持つ場合，d_i, d_j は同じクラスタに所属している可能性が高いということができる．この非類似度による重み付けは d_i, d_j が同一クラスタに含まれることの妥当性を評価しており，クラスタリングにおける分類能力の向上に寄与すると考えられる．

また，式 (2.7) を展開すると，式 (2.9) が得られる．$C_{S \setminus S_r}$ はデータセット全体から d_i, d_j を含むクラスタ r の集合を除いたものの平均を表す．

$$\begin{aligned}
&\mathrm{MVS}(d_i, d_j \mid d_i, d_j \in S_r) \\
&\qquad = d_i^{\mathrm{T}} d_j - d_i^{\mathrm{T}} C_{S \setminus S_r} - d_j^{\mathrm{T}} C_{S \setminus S_r} + 1 \qquad (2.9)
\end{aligned}$$

d_i に対して d_j よりも遠いデータ d_l がある時，$\mathrm{MVS}(d_i, d_j)$ と $\mathrm{MVS}(d_i, d_l)$ の関係から，式 (2.10) が成り立つ．

$$\begin{aligned} &\mathrm{MVS}(d_i, d_j) > \mathrm{MVS}(d_i, d_l) \\ \Leftrightarrow\ &d_i^{\mathrm{T}} d_j - d_j^{\mathrm{T}} C_{S \setminus S_r} > d_i^{\mathrm{T}} d_l - d_l^{\mathrm{T}} C_{S \setminus S_r} \end{aligned} \tag{2.10}$$

もし，単純な cosine 類似度では d_j よりも d_l のほうが d_i に近いと評価した場合 ($d_i^{\mathrm{T}} d_j \leq d_i^{\mathrm{T}} d_l$) でも，$d_l$ が d_j よりもクラスタ外の平均と大きく近い場合にクラスタ外との非類似度により，より同一のクラスタに含まれそうな d_j との類似度を大きく評価できる．つまり，MVS は単純な cosine 類似度に，評価を強固にする項を付け加えたものであると言える．

式 (2.7) からわかるように，MVS は最大 $n-2$ 回の内積の算術平均である．そのため，MVS の計算量は $O(n)$ であり，これは単純な cosine 類似度の約 n 倍の計算量となる．

Nguyen らは，この MVS に基づく目的関数を用いた非階層クラスタリング手法である MVS クラスタリング (MVSC) を提案した．[2] では 2 種類の異なる目的関数を考案した．1 つ目の目的関数は式 (2.11) に示す全てのクラスタにおける同一クラスタメンバ間の MVS の総和である．また，この目的関数を用いたクラスタリングアルゴリズムを MVSC-I_R として提案している．

$$F = \sum_{r=1}^{K} n_r \left[\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} Sim(d_i, d_j) \right] \tag{2.11}$$

2 つ目の目的関数は式 (2.12) に示す全てのクラスタにおけるクラスタメンバとクラスタ重心との MVS の総和である．また，この目的関数を用いたアルゴリズムを MVSC-I_V として提案している．

$$G = \sum_{r=1}^{K} \frac{1}{n - n_r} \sum_{d_i \in S_r} Sim\left(d_i, \frac{C_r}{\parallel C_r \parallel} \right) \tag{2.12}$$

これらの手法において目的関数の最適化は Incremental Optimization Algorithm により，ある 1 データが現在と異なる他クラスタへ移動した場合の目的関数の差分を計算し，目的関数を最大化させるクラスタへの移動を目的関数が収束するまで繰り返すことで実現される．実データを用いた評価実験において，これらの手法は cosine 類似度を用いた非階層クラスタリング手法である Spk-means[1] 等の既存

手法に対して優位性を示した．さらに，Spk-means のクラスタリング結果を初期値とした場合に，その結果を MVSC によって改善できることを示した．

また，MVS を用いた分割型階層クラスタリングは Jayaprada らによって提案されており，この手法では MVS を用いた Bisecting Incremental k-means を再帰的に行うことで top-down による階層的なクラスタ分割を実現している [5]．

2.4　Multi-View クラスタリング

本節では，Bickel らが提唱する Multi-View クラスタリング [6] を紹介する．Multi-View クラスタリングは入力データ集合 X が，異なる特徴量に基づく独立な部分集合 $X^{(1)}, \ldots, X^{(M)}$ からなると想定してクラスタリングを行う手法である．この手法では M 個の各 view がクラスタリングにおいて異なる重要度をもつという仮定のもと，重要度の高い view に大きな影響力を持たせるような重みを学習させることでクラスタリング精度の向上を実現した．ここでは，扱われるデータが異なる特徴量による部分集合によって表現されていることを Multi-View と称している．この Multi-View は，同一のデータを複数の基準点から評価する [2] 及び本研究における多観点とは異なるものである．

2.5　凝集型階層クラスタリング

凝集型階層クラスタリングは，クラスタ数が $K = n$ 個で全てのデータが異なるクラスタに所属する状態からクラスタ数 n が 1 になるまで最近傍の 2 つのクラスタのマージを繰り返すことで階層的なクラスタ分割構造を得る手法である．

階層クラスタリングは，既存のクラスタ a, b からマージにより新しいクラスタ c を作成したとき，そのクラスタ c とその他のクラスタ k に対してのクラスタ間の類似度 Sim_{kc} を求める必要がある．その定義には様々な種類があり，定義ごとに階層クラスタリングの結果は異なるものになる．いずれの方法においても初期状態では，全てのクラスタが単一のデータからなるため，クラスタ間類似度も 2 データ間の類似度に従う．最も単純な手法として単リンク及び完全リンクが挙げられる．単リンク法では，k, a 間の類似度と k, b 間の類似度の大きい方を k, c 間の類似度として採用する．逆に完全リンクでは類似度の小さい方を採用する．群平均法では，k, c の各クラスタメンバ対の類似度の平均をクラスタ間類似度としている．また，重心法は k と c のクラスタ重心間の類似度をクラスタ間類似度としている．

このほかにもメジアン法やウォード法がある．中でも外れ値に強く実用的であることから群平均法は広く用いられている．階層クラスタリングの概略をアルゴリズム 1 に示す．

2.5.1　群平均法

群平均法はクラスタ間類似度を，互いのクラスタメンバ間の類似度の平均とする手法である．クラスタ a, b をマージして新しくできたクラスタ c とその他のクラスタ k との類似度 Sim_{kc} は，式 (2.13) のように表される．

$$Sim_{kc} = \frac{1}{n_k n_c} \sum_{d_i \in S_k} \sum_{d_j \in S_c} Sim(d_i, d_j) \tag{2.13}$$

また，式 (2.14) のように事前に計算されているマージ前のクラスタ間類似度を用いることで，クラスタサイズに基づく重み付き平均で表現することができ，類似度の計算を $O(1)$ でおこなうことが可能である．

$$Sim_{kc} = \frac{n_a}{n_c} Sim_{ka} + \frac{n_b}{n_c} Sim_{kb} \tag{2.14}$$

2.5.2　階層クラスタリングの計算量

アルゴリズム 1 階層クラスタリング

Input: データセット $S = \{d_1, \ldots, d_n\}$

1: $n \times n$ の類似度行列の初期化
2: **while** クラスタ数 > 1 **do**
3:　最も類似度の高いクラスタ a, b を探す
4:　a, b をマージしてクラスタ c を作る
5:　**for** $k \leftarrow c$ 以外のクラスタ **do**
6:　　クラスタ c と k の間の類似度を更新

Output: 階層的なクラスタ構造

ここでは，アルゴリズム 1 を用いて階層クラスタリングにかかる計算量について示す．1 行目では全てのデータ対の間の類似度を持つ $n \times n$ の類似度行列の初期化をおこなう．cosine 類似度は m 次元ベクトルの内積であることから $O(m)$ で計算可能で，類似度行列の初期化の計算量は $O(mn^2)$ である．また，3 行目の最近傍

のクラスタ対の探索は，ヒープを用いて $O(n \log n)$ で計算可能である．6 行目の類似度の更新は，式 (2.14) の更新式により $O(1)$ で計算できる．この計算が，新しいクラスタと他の全クラスタとの間で必要であるため，類似度行列の更新の計算量は $O(n)$ である．出力のクラスタ構造は，各ステップでマージされたクラスタ対とその類似度を持つ．

クラスタリング全体で，クラスタのマージは $n-1$ 回行われる．また，1 回のマージ毎に必要な手続きで最も計算量が大きいものは，$O(n \log n)$ での最近傍のクラスタ対の探索である．そのため，一般的な階層クラスタリングは，類似度行列の初期化と合わせて全体で $O(mn^2 + n^2 \log n)$ の計算量で実行可能である．

2.6　k-means

k-means は非階層クラスタリングの代表的な手法の 1 つである [3]．k-means は，クラスタの平均に基づいてデータ集合を指定した K 個のクラスタに分割する手法であるためこのように呼ばれる．式 (2.15) に k-means において最小化すべき目的関数を示す．

$$Obj(S) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \parallel d_i - \mu_k \parallel^2 \tag{2.15}$$

この式中において，μ_k はクラスタ k におけるクラスタ重心である．また，r_{ik} はデータ d_i がクラスタ k に所属するときに 1，そうでないとき 0 を持つ．よって r_i は 1-of-K 符号化によってデータ d_i の所属クラスタを表現するものである．式 (2.15) より，k-means では K 個のクラスタの重心とクラスタメンバとのユークリッド距離を最小化することで最適なクラスタ分割をおこなっていることがわかる．k-means では，以下のアルゴリズム 2 に示す手続きによって目的関数の最小化をおこなう．

k-means における反復中の処理は大きく次の 2 つに分割することができる．

- データ所属に基づく各クラスタの重心の再計算
- すべてのデータの最も近い重心を持つクラスタへの所属

重心の再計算は目的関数をパラメタ μ に関して最小化する手続きである．また，クラスタへの所属は式 (2.15) の目的関数をパラメタ r について最小化する手続きである．

一般的に，各データの初期状態におけるクラスタ所属は乱数を用いて設定をおこなう．ただし，k-means はこのクラスタ所属の初期化によってクラスタリングの

アルゴリズム 2 k-means

Input: $S = \{d_1, \ldots, d_n\}, r = \{r_{ik} \mid i = 1, \ldots, n\ k = 1, \ldots, K\}, \epsilon$

1: **while** $\Delta Obj > \epsilon$ **do**
2:　**for** $k = 1, \ldots, K$ **do**
3:　　$\mu_k = \frac{1}{n_k} \sum_{d_i \in S_k} d_i$
4:　**for** $i \leftarrow 1, \ldots, n$ **do**
5:　　**for** $k \leftarrow 1, \ldots, K$ **do**
6:　　$dist(d_i, \mu_k) = \| d_i - \mu_k \|^2$
7:　　$r_{ik} = \begin{cases} 1 & (k = \arg\min_k dist(d_i, \mu_k)) \\ 0 & (\text{otherwise}) \end{cases}$
8:　ΔObj の計算

Output: クラスタ所属 r

結果が変化する．これは初期値依存性と呼ばれ k-means の抱える欠点の 1 つである．そのため，k-means では複数回のクラスタリング結果の内から最も優れた結果を採用するといった方法が用いられる．

2.6.1　k-means の計算量評価

アルゴリズム 2 に示した通り，k-means では目的関数の変動が収束するまでクラスタ分割の更新を繰り返す．ここでは，目的関数の収束に必要な繰り返し回数はせいぜい定数回であるとする．各繰り返しの中において，まずは各データを最も距離の近いクラスタに所属させる手続きの計算量を考える．この手続きでは n 個のデータに対して，K 種類のクラスタ重心とのユークリッド距離を計算している．m 次元のデータ対に対して式 (2.16) に示す 1 回の距離計算には $O(m)$ かかる．

$$\| x - y \| = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{2.16}$$

この式中で x_i は x の i 次元目の要素の値である．そのためクラスタ所属の最適化に必要な計算量は $O(mnK)$ である．

続いて，K 種類のクラスタについて重心を計算する手続きに必要な計算量について考える．あるクラスタ k に関して重心 μ_k は式 (2.17) のように m 次元のクラ

スタメンバの平均ベクトルで求められる．

$$\mu_k = \frac{\sum_{d_i \in S_k} d_i}{n_k} \tag{2.17}$$

よってクラスタ k の重心を求める計算量は $O(mn_k)$ である．この計算をすべてのクラスタについておこなうが，ここで $n = n_1 + \cdots + n_K$ であることを考慮すると $O(m(n_1 + \cdots + n_K)) = O(mn)$ である．よって，K 種類のクラスタ重心の計算について必要な計算量は $O(mn)$ である．

そのため，k-means の計算量は目的関数の収束が定数回の繰り返しで達成されるならば $O(mnK)$ である．

第3章

多観点類似度を用いた階層クラスタリング

本節では，cosine 類似度に関する MVS を適用した階層クラスタリング手法について論じる．Nguyen らの研究により，MVS が分割クラスタリングにおいて，cosine 類似度を用いたクラスタリングの結果を改善することが示されている．この cosine 類似度に対する MVS の優位性は，階層クラスタリングにおいても同様に現れることが想定できる．そこで，類似度指標に MVS を用いた階層クラスタリング手法を提案する．MVS を階層クラスタリングに適用することで，cosine 類似度よりも高い分類精度に基づきながらも，階層的な分類構造を得ることができる．また，階層クラスタリングにおけるクラスタ間類似度の定義には一般に広く利用されている群平均法を用いる．

3.1 MVS を導入したクラスタ間類似度

MVS を階層クラスタリングに適用する場合，階層クラスタリングにおけるクラスタ間類似度を MVS に準拠した定義に変更する必要がある．群平均法において，クラスタ間類似度は式 (2.13) のように求められていた．この式中の類似度を MVS にすることで，MVS によるクラスタ間類似度を式 (3.1) ように表すことができる．

$$Sim_{ab} = \frac{1}{n_a n_b} \sum_{d_i \in S_a} \sum_{d_j \in S_b} \mathrm{MVS}(d_i, d_j) \tag{3.1}$$

この式 (3.1) の中では，異なるクラスタ a, b にそれぞれ属する d_i, d_j について MVS を計算している．しかし，式 (2.7) からわかるように，MVS は同一クラスタ内の

データ間に対してのみ定義されているため，異なるクラスタのデータ間について計算することができない．そこで，d_i, d_j が異なるクラスタ a, b に所属するときの MVS を，式 (3.2) のように a, b がマージされている状態を仮定して計算することとする．a, b がマージされている時，そのクラスタのデータ集合は $S_a \cup S_b$ で得られる．また，クラスタリングにおいて各データは唯一のクラスタに属することから，マージされたクラスタのデータ数は $n_a + n_b$ になる．

$$\mathrm{MVS}(d_i, d_j \mid d_i \in S_a, d_j \in S_b) = \frac{1}{n - (n_a + n_b)} \sum_{d_h \in S \setminus (S_a \cup S_b)} (d_i - d_h)^{\mathrm{T}} (d_j - d_h) \quad (3.2)$$

3.2　階層クラスタリングにおいて変更が必要となる処理

3.1 節では MVS を用いた場合のクラスタ間類似度について述べた．実際に階層クラスタリングに MVS を導入するには，その手続きの内類似度計算が行われる部分について変更が必要となる．1 つ目は，初期状態における各クラスタ間の類似度による行列を求める手続きである．もう 1 つは，クラスタをマージして新しいクラスタを作ったときに必要となる，新しいクラスタとその他のクラスタとの間の類似度計算による類似度行列の更新である．

3.3　MVS を導入した階層クラスタリングの単純手法

本節では，MVS を単純に群平均法に導入すると計算量が増加してしまうことを述べる．本研究の提案手法については，3.4 節で述べるが，これは計算量の増加を抑制するように洗練したものである．Cosine 類似度を用いた場合，階層クラスタリング全体の計算量は 2.5.2 節で論じたように $O(mn^2 + n^2 \log n)$ である．

3.3.1　類似度行列の初期化

このうち，類似度行列の初期化部分の計算量は $O(mn^2)$ である．これは，1 回あたり $O(m)$ の cosine 類似度の計算が，$n \times n$ の類似度行列の各要素に対して必要となるためである．ここで，cosine 類似度を計算量が $O(mn)$ である MVS に変更した場合，$O(mn)$ の計算を $O(n^2)$ 回おこなうため類似度行列初期化の計算量が $O(mn^3)$ になる．これは $O(mn^2 + n^2 \log n)$ よりも大きい計算量であるため，階層

クラスタリング全体の計算量を増加させるといえる．

3.3.2　マージ後の類似度行列の更新

階層クラスタリングでは，n 個のクラスタを 1 つにするまでマージを繰り返す手続きを $O(n^2 \log n)$ で実現している．また，マージ 1 回ごとの計算量は $O(n \log n)$ である．この部分では，マージによって作られた新しいクラスタと，最大で $n-2$ 個あるその他のクラスタの間のクラスタ間類似度を式 (2.14) により $O(1)$ で計算することで，マージによる類似度行列の更新を $O(n)$ で計算可能にしている．ただし，この $O(1)$ 更新式は，クラスタリング中における各データのクラスタ所属の変化に対して不変であるような類似度にのみ用いることができる．cosine 類似度では，その値が類似度を計算する 2 つのデータのみで決定されるためこの更新式を用いることができた．しかし，MVS は，クラスタの構成により基準点として採用するデータ集合が変化するため，クラスタリング中にその値が変化する場合がある．例えば，クラスタ a のあるデータ d_i, d_j の類似度は

$$\mathrm{MVS}(d_i, d_j) = \frac{1}{n - n_a} \sum_{d_h \in S \setminus S_a} (d_i - d_h)^{\mathrm{T}} (d_j - d_h) \tag{3.3}$$

であった．ここで，このクラスタ a が他のクラスタ b とマージされたなら，その後の d_i, d_j 間の類似度は

$$\mathrm{MVS}(d_i, d_j) = \frac{1}{n - (n_a + n_b)} \sum_{d_h \in S \setminus (S_a \cup S_b)} (d_i - d_h)^{\mathrm{T}} (d_j - d_h) \tag{3.4}$$

であるため，マージ前とは類似度が異なる．そのため，MVS では式 (2.14) の更新式を用いてクラスタ間類似度を更新することができない．

MVS を用いた場合の群平均法におけるクラスタ間類似度の計算量は式 (3.1) に示したとおり $O(mn^3)$ であり，この式を用いてマージ後のクラスタ間類似度を更新した場合，1 回のマージ毎の類似度行列の更新に必要な計算量は $O(mn^4)$ になる．また，階層クラスタリングではマージが $n-1$ 回繰り返されることから，類似度行列の更新に対して単純に MVS を導入した場合，階層クラスタリング全体の計算量は $O(mn^5)$ になる．これは $O(mn^2 + n^2 \log n)$ に対して非常に大きいため，単純に MVS を導入することで階層クラスタリングの計算量を大幅に悪化させてしまうことがわかる．

3.4　MVS を導入した階層クラスタリングの高速化手法

前節で示したとおり，MVS を単純に階層クラスタリングに適用した場合，計算量が $O(mn^5)$ と非常に大きくなる．本節では，MVS を導入した階層クラスタリングの計算量を cosine 類似度を用いた場合の $O(mn^2 + n^2 \log n)$ と同等に低減するために，類似度の計算が必要となる手続きに対する高速化手法を提案する．

3.4.1　導入した類似度行列の初期化

cosine 類似度を用いた場合，類似度行列の初期化の計算量は $O(mn^2)$ であった．そのため，ここでは MVS を用いた類似度行列を計算量 $O(mn^2)$ で計算する手法を提案する．

階層クラスタリングにおいて，最初の状態では全てのデータが別々のクラスタを形成している．よってクラスタ数は n 個であり，各クラスタはデータを 1 つのみ保有している．そのため任意のクラスタ $c = 1, \ldots, n$ についてデータ集合 S_c 及びデータ数 n は，

$$S_c = d_c, \quad n_c = 1$$

である．式 (3.1) の MVS を適用した群平均法におけるクラスタ間類似度は，これらを考慮することで式 (3.5) のように表現できる．

$$\begin{aligned} Sim_{ij} &= \frac{1}{n-2} \sum_{d_h \in S \setminus d_i \setminus d_j} (d_i^{\mathrm{T}} d_j - d_i^{\mathrm{T}} d_h - d_j^{\mathrm{T}} d_h + 1) \\ &= \frac{1}{n-2} \left\{ (n-2) d_i^{\mathrm{T}} d_j - d_i^{\mathrm{T}} (D - d_i - d_j) - d_j^{\mathrm{T}} (D - d_i - d_j) + (n-2) \right\} \\ &= \frac{1}{n-2} (n d_i^{\mathrm{T}} d_j - d_i^{\mathrm{T}} D - d_j^{\mathrm{T}} D + n) \end{aligned} \tag{3.5}$$

ここで，D は全データの総和ベクトル $D = \sum_{d_i \in S} d_i$ である．D の計算には $O(mn)$ かかるが，D を事前に計算している場合，式 (3.5) は $O(m)$ の内積計算 3 回の和となり，この式による MVS の計算量は $O(m)$ である．この計算を $n \times n$ 類似度行列に対して用いた場合，初期化全体の計算量は $O(mn^2)$ であり，cosine 類似度を用いた場合と同等になる．式 (3.5) を用いるには前述のとおり，$O(mn)$ による D の事前計算が必要となるが，$n \times n$ 行列全体に対して 1 度だけ計算すればよいので初期化全体の計算量には大きく影響しない．

式 (3.5) による MVS を厳密に cosine 類似度と比較すると，内積の計算回数は 3 倍になる．そのため，MVS を用いた類似度行列の初期化の計算量が cosine 類似度

に対して約 3 倍になってしまう．次元数 m の大きいデータに対するクラスタリングでは，全体の計算量の中でも初期化の部分は大きい影響力を持つため，3 倍という計算回数の増加は無視できない．そこで，類似度行列の初期化全体において式 (3.5) 中に繰り返し出現する共通項の事前計算により，一度の類似度計算あたりの内積計算の回数を 1 回にする方法を示す．式 (3.5) において，第 2,3 項は 1 つのデータ点と D の内積である．この $d_i^{\mathrm{T}} D (i = 1, \ldots, n)$ は高々n 種類しか存在せず，類似度行列の初期化における n^2 回の計算の中では同じものが n 回ずつ繰り返し使われている．そのため，n 種類あるこの内積計算を事前に $O(mn)$ でおこなうことで，式 (3.5) における内積計算回数を cosine 類似度と同様の 1 回にすることができる．

アルゴリズム 3 に MVS を用いた類似度行列の初期化の流れを示す．MVS の類似度行列を求めるためには，

- D の事前計算
- 事前計算した D を用いた $d_i^{\mathrm{T}} D (i = 1, \ldots, n)$ の事前計算
- 事前計算した $d_i^{\mathrm{T}} D (i = 1, \ldots, n)$ を用いた MVS 類似度行列の計算

の 3 つの手続きが必要である．これらを組み合わせると計算量は $O(mn+mn+mn^2)$ である．そのため MVS の類似度行列の初期化の計算量は，cosine 類似度を用いた場合の計算量と同等の $O(mn^2)$ であると言える．

アルゴリズム 3 MVS による類似度行列の初期化

Input: データセット $S = \{d_1, \ldots, d_n\}$

1: $D \leftarrow \sum_{i=1}^{n} d_i$
2: **for** $i \leftarrow 1, \ldots, n$ **do**
3:　$d_i^{\mathrm{T}} D$ の事前計算
4: **for** $i \leftarrow 1, \ldots, n$ **do**
5:　**for** $j \leftarrow 1, \ldots, n$ **do**
6:　$Sim_{ij} \leftarrow \frac{1}{n-2}(n d_i^{\mathrm{T}} d_j - d_i^{\mathrm{T}} D - d_j^{\mathrm{T}} D + n)$

Output: $n \times n$ 類似度行列 Sim

$m > n$ の場合の類似度行列の高速化

ここでは，次元数 $m > n$ の条件で有効な高速化手法を説明する．ここでは式 (3.5) のなかで，第 1 項 $d_i^{\mathrm{T}} d_j$ は d_i, d_j の cosine 類似度であることに注目する．また，

任意のデータ点 d_i と全データの総和ベクトル D との内積は，式 (3.6) ように表せる．

$$d_i^{\mathrm{T}} D = \sum_{j=1}^{n} d_i^{\mathrm{T}} d_j \tag{3.6}$$

式 (3.6) の左辺は m 次元ベクトルの内積であるが，n 個のスカラー値の総和として $O(n)$ で事前計算できる．さらに，式 (3.5) の第 1〜3 項は，あらかじめ $O(mn^2)$ で cosine 類似度行列が計算されている場合，内積の計算を必要とすることなくスカラー値の和から $O(1)$ で計算できる．

アルゴリズム 4 に cosine 類似度の事前計算を用いた MVS の類似度行列の初期化の流れを示す．この cosine 類似度行列の事前計算を用いた初期化法の計算量は，

- cosine 類似度行列の事前計算
- cosine 類似度行列を用いた $d_i^{\mathrm{T}} D (i = 1, \ldots, n)$ の事前計算
- 事前計算した cosine 類似度及び $d_i^{\mathrm{T}} D (i = 1, \ldots, n)$ を用いた MVS 類似度行列の計算

の手続きの組み合わせより $O(mn^2 + n^2 + n^2) = O(mn^2)$ である．これは前述の初期化法と同等の計算量であるが，データ集合の次元数 m が $m > n$ である場合こちらのほうが高速になる．

アルゴリズム 4 MVS による類似度行列の初期化 ($m > n$ の場合)

Input: データセット $S = \{d_1, \ldots, d_n\}$

1: **for** $i \leftarrow 1, \ldots, n$ **do**
2:　　**for** $j \leftarrow 1, \ldots, n$ **do**
3:　　　　$\mathrm{CS}(d_i, d_j) \leftarrow d_i^{\mathrm{T}} d_j$
4: **for** $i \leftarrow 1, \ldots, n$ **do**
5:　　$d_i^{\mathrm{T}} D = \sum_{j=1}^{n} \mathrm{CS}(d_i, d_j)$
6: **for** $i \leftarrow 1, \ldots, n$ **do**
7:　　**for** $j \leftarrow 1, \ldots, n$ **do**
8:　　　　$Sim_{ij} \leftarrow \frac{1}{n-2}(n\mathrm{CS}(d_i, d_j) - d_i^{\mathrm{T}} D - d_j^{\mathrm{T}} D + n)$

Output: $n \times n$ 類似度行列 Sim

3.4.2　マージ後の類似度行列の更新式

3.3 節では，式 (2.14) を用いたマージ後のクラスタ間類似度の更新式は MVS に応用できないことを示した．そのためここでは，マージ前後で類似度の値が変動する MVS の特性を考慮しながら，式 (2.14) のようなマージ前のクラスタ間類似度を用いた高速な群平均法の更新式を導出する．

MVS を適用した群平均法による更新をおこなう階層クラスタリングにおいて，クラスタ a, b をマージして新しくクラスタ c を作った時，式 (2.13) における $Sim(d_i, d_j)$ に，式 (2.7) に示す MVS を用いると次の式 (3.7) のようになる．

$$Sim_{kc} = \frac{1}{n_k n_c (n - n_k - n_c)} \sum_{d_k \in S_k} \sum_{d_c \in S_c} \sum_{d_h \in S \setminus S_k \setminus S_c} (d_k - d_h)^{\mathrm{T}} (d_c - d_h) \tag{3.7}$$

また，Sim_{ka}, Sim_{kb} も同様に式 (3.8)，式 (3.9) のようになる．

$$Sim_{ka} = \frac{1}{n_k n_a (n - n_k - n_a)} \sum_{d_k \in S_k} \sum_{d_a \in S_a} \sum_{d_h \in S \setminus S_k \setminus S_a} (d_k - d_h)^{\mathrm{T}} (d_a - d_h) \tag{3.8}$$

$$Sim_{kb} = \frac{1}{n_k n_b (n - n_k - n_b)} \sum_{d_k \in S_k} \sum_{d_b \in S_b} \sum_{d_h \in S \setminus S_k \setminus S_b} (d_k - d_h)^{\mathrm{T}} (d_b - d_h) \tag{3.9}$$

ここで，クラスタ c がクラスタ a, b のマージによるものであることから，式 (3.7) は次の式 (3.10) のように表すことができる．

$$Sim_{kc} = \frac{1}{n_k (n_a + n_b)(n - n_k - n_a - n_b)} \sum_{d_k \in S_k} \left\{ \sum_{d_a \in S_a} \sum_{d_h \in S \setminus S_k \setminus S_a \setminus S_b} (d_k - d_h)^{\mathrm{T}} (d_a - d_h) + \sum_{d_b \in S_b} \sum_{d_h \in S \setminus S_k \setminus S_a \setminus S_b} (d_k - d_h)^{\mathrm{T}} (d_b - d_h) \right\} \tag{3.10}$$

そのため Sim_{kc} は Sim_{ka}, Sim_{kb} を用いて次の式 (3.11) のように表現できる．

$$\begin{aligned} Sim_{kc} &= \frac{n_a(n-n_k-n_a)}{(n_a+n_b)(n-n_k-n_a-n_b)}Sim_{ka} \\ &\quad + \frac{n_b(n-n_k-n_b)}{(n_a+n_b)(n-n_k-n_a-n_b)}Sim_{kb} \\ &\quad - \frac{1}{n_k(n_a+n_b)(n-n_k-n_a-n_b)} \sum_{d_k\in S_k}\sum_{d_a\in S_a}\sum_{d_h\in S_b}(d_k-d_h)^{\mathrm{T}}(d_a-d_h) \\ &\quad - \frac{1}{n_k(n_a+n_b)(n-n_k-n_a-n_b)} \sum_{d_k\in S_k}\sum_{d_b\in S_b}\sum_{d_h\in S_a}(d_k-d_h)^{\mathrm{T}}(d_b-d_h) \end{aligned} \tag{3.11}$$

ここで，各クラスタの総和ベクトル $D_a = \sum_{d_i\in S_a} d_i$ を用いることで第 3 項について，

$$\begin{aligned} & \frac{1}{n_k(n_a+n_b)(n-n_k-n_a-n_b)} \sum_{d_k\in S_k}\sum_{d_a\in S_a}\sum_{d_h\in S_b}(d_k-d_h)^{\mathrm{T}}(d_a-d_h) \\ = & \frac{1}{n_k(n_a+n_b)(n-n_k-n_a-n_b)} \left(n_b D_k^{\mathrm{T}}D_a - n_a D_k^{\mathrm{T}}D_b - n_k D_a^{\mathrm{T}}D_b + n_k n_a n_b\right) \end{aligned}$$

また，同様に第 4 項についても，

$$\begin{aligned} & \frac{1}{n_k(n_a+n_b)(n-n_k-n_a-n_b)} \sum_{d_k\in S_k}\sum_{d_b\in S_b}\sum_{d_h\in S_a}(d_k-d_h)^{\mathrm{T}}(d_b-d_h) \\ = & \frac{1}{n_k(n_a+n_b)(n-n_k-n_a-n_b)} \left(n_a D_k^{\mathrm{T}}D_b - n_b D_k^{\mathrm{T}}D_a - n_k D_b^{\mathrm{T}}D_a + n_k n_a n_b\right) \end{aligned}$$

のように表すことができる．よってマージ後のクラスタ間類似度の更新式を，マージ前の Sim_{ka}, Sim_{kb} を用いて以下の式 (3.12) のように導出できる．

$$\begin{aligned} Sim_{kc} &= \frac{1}{(n_a+n_b)(n-n_k-n_a-n_b)} \\ &\quad \Big\{ n_a(n-n_k-n_a)Sim_{ka} \\ &\quad\quad + n_b(n-n_k-n_b)Sim_{kb} \\ &\quad\quad + 2\left(D_a^{\mathrm{T}}D_b - n_a n_b\right) \Big\} \end{aligned} \tag{3.12}$$

この式は第 3 項に m 次元ベクトルの内積を含むため計算量は $O(m)$ である．これは式 (2.14) を用いた cosine 類似度の更新の計算量 $O(1)$ よりも大きくなる．

式 (3.12) の計算には，D_a, D_b, n_a, n_b, n_k が必要である．そのため，クラスタリングの過程において最近傍のクラスタ対 a, b のマージによって新しいクラスタ c が

作られたとき，そのクラスタに関して D_c, n_c を更新する操作が必要になる．階層クラスタリングの最初の状態では全てのデータが異なるクラスタに所属していることから，任意のクラスタ $c = 1, \ldots, n$ について，

$$D_c = d_c, \quad n_c = 1$$

である．またクラスタ c が，既存のクラスタ a, b のマージによって新しく作られたとき，

$$D_c = D_a + D_b, \quad n_c = n_a + n_b$$

であり，計算量はそれぞれ $O(m)$ と $O(1)$ である．

階層クラスタリングでは，クラスタが 1 つになるまで最近傍のクラスタ対のマージを繰り返すが，そのマージ 1 回の中でクラスタ間類似度行列の更新には，最大で $n-2$ 回の類似度計算が必要である．よって式 (3.12) を用いた場合，マージ 1 回につきかかる計算量は $O(mn)$ である．また，更新式に必要なクラスタの総和ベクトルの更新は，1 回のマージ毎に 1 度のみおこなうため $O(m)$ なので，類似度行列の更新に必要な計算量は $O(mn)$ である．

ここで，式 (3.12) 中の $D_a^{\mathrm{T}} D_b$ に注目する．この D_a 及び D_b はクラスタ c を構成する 2 クラスタ a, b の総和ベクトルであった．この $D_a^{\mathrm{T}} D_b$ を事前計算することで，類似度行列の更新を高速化することができる．$D_a^{\mathrm{T}} D_b$ が事前計算されているならば，式 (3.12) は $O(1)$ で計算可能となるため，1 回のマージでの計算量は $O(n)$ になる．よって，最近傍クラスタ a, b が決定された時点で $D_a^{\mathrm{T}} D_b$ の計算をあらかじめ計算量 $O(m)$ でおこなうことで，マージ 1 回毎の類似度行列の更新を $O(mn)$ から $O(m+n)$ に減少させることができる．

$m > n$ の場合には，式 (3.12) 中の $D_a^{\mathrm{T}} D_b$ に注目する．クラスタリングの初期状態において，$D_i = d_i (i = 1, \ldots, n)$ であることから，$D_a^{\mathrm{T}} D_b = d_a^{\mathrm{T}} d_b$ である．よって $D_a^{\mathrm{T}} D_b (a, b = 1, \ldots, n)$ を $n \times n$ 行列で表現した場合，これは cosine 類似度行列と等価であることがわかる．さらに，クラスタ a, b のマージによる新しいクラスタ c

に関してその他のクラスタ k との $D_c^{\mathrm{T}} D_k$ は

$$\begin{aligned} D_c^{\mathrm{T}} D_k &= \sum_{d_c \in S_c} \sum_{d_k \in S_k} d_c^{\mathrm{T}} d_k \\ &= \sum_{d_c \in (S_a \cup S_b)} \sum_{d_k \in S_k} d_c^{\mathrm{T}} d_k \\ &= \sum_{d_a \in S_a} \sum_{d_k \in S_k} d_a^{\mathrm{T}} d_k + \sum_{d_b \in S_b} \sum_{d_k \in S_k} d_b^{\mathrm{T}} d_k \\ &= D_a^{\mathrm{T}} D_k + D_b^{\mathrm{T}} D_k \end{aligned}$$

であるため，スカラー値の和により $O(1)$ 更新することができる．一度のマージ毎に新しいクラスタ c に対して最大 $n-2$ 個のクラスタ間で $D_c^{\mathrm{T}} D_k$(k は c 以外の全クラスタ) を計算するため，更新に必要な計算量は $O(n)$ である．よって cosine 類似度行列が事前に計算されているならば，マージ 1 回毎の類似度行列の更新を $O(n+n) = O(n)$ でおこなうことができる．この事前計算した cosine 類似度行列を用いた手法によって，類似度行列の更新においてもデータセットの次元数 m が $m > n$ の場合に更なる高速化を望むことができる．

3.5　MVS を導入した階層クラスタリングの計算量

本節では，3.4.1 節及び 3.4.2 節にて提案した MVS の導入を用いた階層クラスタリングが，cosine 類似度を用いた階層クラスタリングの計算量 $O(mn^2 + n^2 \log n)$ と同等で実現されていることを示す．類似度行列の初期化は，MVS を導入した場合，3.4.1 節に示したとおり，計算量 $O(mn^2)$ であった．最近傍クラスタのマージを繰り返す手続き全体は，

- ヒープを用いた最近傍クラスタ対 a, b の検索
- 類似度行列の更新
- マージ後のクラスタ c とその他のクラスタ k との $D_c^{\mathrm{T}} D_k$ の更新

の組み合わせの $n-1$ 回の繰り返しからなるため，その計算量は $O(n(n \log n + n + n)) = O(n^2 + n^2 + n^2 \log n)$ である．よって階層クラスタリング全体の計算量は，類似度行列の初期化と最近傍クラスタ対のマージを繰り返す手続きの組み合わせから $O(mn^2 + n^2 + n^2 + n^2 \log n) = O(mn^2 + n^2 \log n)$ になる．これは，cosine 類似度を用いた階層クラスタリングと同等の計算量であることがわかる．よって，3.4.1

節及び 3.4.2 節にて示した手法により，階層クラスタリングの計算量を増加させることなく MVS の導入を実現できることがわかる．

また，3.4.2 節では，$D_a^{\mathrm{T}} D_b$ の事前計算によってマージ 1 回毎における類似度行列の更新に必要な計算量を $O(mn)$ から $O(n)$ に減少させた．この操作を行わない場合，階層クラスタリング全体における類似度行列の更新の計算量は $O(mn^2)$ である．そのため，階層クラスタリング全体の計算量は $O(mn^2 + mn^2 + n^2 \log n)$ である．これは $O(mn^2 + n^2 \log n)$ と等価であるため概算では計算量に変化はない．しかし，$D_a^{\mathrm{T}} D_b$ の事前計算をおこなった場合に対して，実計算上で影響の大きい $O(mn^2)$ の手続きの回数が多くなっていることがわかる．よって，$D_a^{\mathrm{T}} D_b$ の事前計算は実計算上において計算処理の減少に貢献することができると考えられる．

3.6　クラスタサイズの均衡化

MVS を用いた階層クラスタリングでは，クラスタサイズの均衡性を容易に制御することができる．データ集合の解析の観点から，クラスタリングの結果における各クラスタサイズには大きな偏りがないことが望まれる．MVS を用いた場合では，3.4.1 節に示した類似度行列の初期化において式 (3.13) のように任意の定数 λ を減算することで，この定数項に従ってクラスタサイズの均衡性を制御することができる．

$$\widehat{\mathrm{MVS}}(d_i, d_j \mid \lambda) = \mathrm{MVS}(d_i, d_j) - \lambda \tag{3.13}$$

このように類似度行列の初期化に定数の減算を加えた場合，クラスタリング中にマージが発生したとき，マージ後のクラスタについてこの定数の影響が大きくなる．そのためクラスタ間類似度を更新するとき，より多くのマージによって構成されたクラスタとのクラスタ間類似度は小さく評価されやすくなる．この効果により極端にサイズの大きいクラスタの生成が妨げられ，均衡性の高いクラスタ分割が可能になる．

3.7　評価実験

ここまででは，分割クラスタリング問題において cosine 類似度よりも高い分類精度を示す MVS を階層クラスタリングに対して導入する方法及び，MVS を導入した場合にも階層クラスタリング全体の計算量を $O(mn^2 + n^2 \log n)$ に保つための

高速化手法を提案した．本節では，提案手法による MVS を導入した階層クラスタリングが，cosine 類似度を用いた既存手法に対して優位性があることを実データに対する数値実験により示す．この数値実験では，文書データセットに対して各手法で階層クラスタリングを行い，データセットのクラス数と同じクラスタ数によるクラスタ分割をおこなった結果の一致度をクラスタリングの精度として評価する．また，階層クラスタリングをおこなう上で計算処理にかかった時間を評価することで，MVS を用いた階層クラスタリングが cosine 類似度に対して同等の処理時間でより高い精度での分類が可能であることを示す．さらに，3.4.1 節及び 3.4.2 節では，計算上の共通部分の事前計算による定数倍の計算回数削減を可能とする手法を示した．これらが実際にクラスタリングをおこなう上で計算時間に与える影響について，事前計算の有無による計算時間の変化に基づいて論じる．また，3.6 節ではクラスタリング結果の均衡化手法について論じた．この手法を導入した MVS を Balanced MVS(BMVS) と呼ぶ．この BMVS を用いた場合のクラスタサイズの均衡性への影響について，MVS との比較実験により評価する．

3.7.1　実験に用いるデータセット及び前処理

実験には，表 3–1 に示す 18 種類のデータセットを用いる．これらのデータセットは，テキストデータマイニングツール CLUTO 上で提供されている [7]．これらは，[2] を含める多数の先行研究で用いられている．

表中において，c はデータセットのもつクラス数，n はデータ数，m は単語数である．データセットはストップワード除去とステミングをおこなう．さらに，その後 99.5%以上の文書データにおいて出現する頻出語及び，1 つの文書データでしか出現しない希少語の除去をおこなう．この頻出語及び希少語はストップワードと同様にデータセット中において文書データの特徴を表現しうる単語ではないためである．これらの前処理の後 TF-IDF を用いた単位特徴ベクトル化をおこなう．

3.7.2　分類精度の評価指標

クラスタリングの分類精度は，データセットのもつ真のクラスラベルとクラスタリング結果によるラベルの一致度から評価する．ここでは，ラベルの一致度に正規化相互情報量 (Normalized Mutual Information:NMI) を用いる．NMI は次の

表 3–1: 実験で使用する文書データセット

データセット	情報源	c	n	m
fbis	TREC	17	2,463	2,000
hitech	TREC	6	2,301	13,170
k1a	WebACE	20	2,340	13,859
k1b	WebACE	6	2,340	13,859
la1	TREC	6	3,204	17,273
la2	TREC	6	3,075	15,211
re0	Reuters	13	1,504	2,886
re1	Reuters	25	1,657	3,758
tr31	TREC	7	927	10,127
reviews	TREC	5	4,096	23,220
wap	WebACE	20	1,560	8,440
la12	TREC	6	6,279	21,604
new3	TREC	44	9,558	36,306
sports	TREC	7	8,580	18,324
tr11	TREC	9	414	6,424
tr12	TREC	8	313	5,799
tr23	TREC	6	204	5,831
tr45	TREC	10	690	8,260

式 (3.14) で表される．

$$\mathrm{NMI}(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} \tag{3.14}$$

ここで $I(X,Y)$ は相互情報量 (Mutual Information:MI) であり，式 (3.15) で表される．また，$\sqrt{H(X)H(Y)}$ による正規化をおこなうことで NMI は 0〜1 の値を取る．

$$I(X,Y) \;=\; H(X) + H(Y) - H(X,Y) \tag{3.15}$$

ここで，エントロピー $H(X)$ は式 (3.16) であることを用いると $I(X,Y)$ は式 (3.17) のように表現できる．

$$H(X) = -\sum_i P(x_i) \log P(x_i) \tag{3.16}$$

$$I(X,Y) = \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)p(y_j)} \tag{3.17}$$

n 個のデータセットが，真のラベルによってクラス分けされた事象を X に当てはめる．その時，データセットから無作為に選んだデータがクラス i のデータである事象 x_i の確率 $P(x_i)$ はクラス i のデータ数 n_i を用いて，

$$P(x_i) = \frac{n_i}{n} \tag{3.18}$$

である．また，同様にデータセットをクラスタリング結果によってクラスタ分割した事象 Y に対して，無作為に選んだデータがクラスタ j に所属する事象 y_j の確率 $P(y_j)$ はクラスタ j のデータ数 n_j を用いて，

$$P(y_j) = \frac{n_j}{n} \tag{3.19}$$

である．さらに，クラス i のデータであり，クラスタ j に所属するデータの数を $n_{i,j}$ とするとき，無作為に選んだデータがクラス i のデータであり, クラスタ j に所属する確率 $P(x_i, y_j)$ は

$$P(x_i, y_j) = \frac{n_{i,j}}{n} \tag{3.20}$$

である．このことから真のクラスラベルとクラスタリング結果によるラベルの一致度としての NMI は式 (3.21) のように表される．

$$\mathrm{NMI} = \frac{\sum_i \sum_j n_{i,j} \log \frac{n \cdot n_{i,j}}{n_i n_j}}{\sqrt{(\sum_i n_i \log \frac{n_i}{n})(\sum_j n_j \log \frac{n_j}{n})}} \tag{3.21}$$

3.7.3　数値実験の実施環境

本実験では実験用のクラスタリング実行プログラムを以下に示す計算機上において動作させて評価をおこなった．

CPU ： Intel® Core™ i7-4790 CPU @ 3.60GHz × 8
メインメモリ ： 16GB (8GB×2)
OS ： Ubuntu 16.04 LTS

本実験では各クラスタリング手法について C++を用いて実験用プログラムを作成した．実行ファイルには GCC 5.4.0 によりコンパイルしたものを用いた．また，コンパイルには最適化オプション-O2 を用いた．

また，実験で取り扱う文書データは非常に次元数が大きいが，一方で多くの次元において重みが 0 であるような疎ベクトルの集合である．m 次元ベクトル x, y の加算と内積はそれぞれ

$$x + y = \{x_1 + y_1, \ldots, x_m + y_m\}, \quad x^{\mathrm{T}} y = \sum_{i=1}^{m} x_i y_i$$

のように計算される．ここで x_i, y_i はそれぞれの i 次元目の要素の値である．この式からわかるように加算においては $x_i = y_i = 0$ である場合 $(x + y)_i = 0$ である．また，内積においては x_i, y_i のどちらかが 0 であるならば i 次元目においての計算結果は全体に影響しない．本実験ではこの性質を用いて加算では $x_i = y_i = 0$ であるとき，内積では $x_i = 0$ または $y_i = 0$ であるときに計算を省略することで高速なベクトル計算をおこなう．

3.7.4　分類精度の評価

表 3–2 左に，クラスタリング結果について，NMI を用いた分類精度の評価実験の結果を示す．この実験では 18 個のデータセットのうち 13 個において cosine 類似度よりも MVS のほうが一致度が高い結果となった．このことから，MVS を用いた階層クラスタリングでは cosine 類似度よりも高い分類精度が得られることがわかった．また，$\lambda = \frac{2}{n-2}$ による BMVS を用いた階層クラスタリングでは re1 を除く 17 個のデータセットについて cosine 類似度よりも高い分類精度を示した．

3.7.5　計算時間の評価

ここでは，MVS を適用した階層型クラスタリングが実データのクラスタリングに用いた場合にも従来手法と同程度の処理時間で実行できることを示す．

表 3–2 右に，処理時間の比較実験の結果を示す．計算処理時間は実行毎に異なる結果が得られることを考慮して実験結果の数値は各条件ごとに 3 回試行した結果の平均を用いた．表中の rate は，MVS の実行時間と cosine 類似度の実行時間の比を表す．

$$\mathrm{rate} = \frac{\text{MVS を用いた場合の処理時間}}{\text{cosine 類似度を用いた場合の処理時間}}$$

この結果から，MVS を用いた場合殆どのデータセットにおいておおよそ 1.01〜1.02 倍程度 cosine 類似度よりも処理時間がかかることがわかる．よって，実データに

表 3–2: 分類精度及び処理時間の評価

データ	分類精度			処理時間 (sec)		
	MVS	BMVS	CS	MVS	CS	rate
fbis	0.570	0.584	0.561	21.246	21.131	1.005
hitech	0.250	0.255	0.059	17.031	17.018	1.001
k1a	0.556	0.556	0.550	17.733	14.416	1.018
k1b	0.710	0.710	0.666	17.654	17.521	1.008
la1	0.383	0.376	0.316	42.956	42.271	1.016
la2	0.374	0.466	0.390	37.135	36.900	1.006
re0	0.312	0.312	0.296	4.427	4.394	1.008
re1	0.540	0.540	0.568	6.165	6.078	1.014
tr31	0.670	0.670	0.527	1.493	1.470	1.016
reviews	0.410	0.420	0.034	85.972	85.096	1.182
wap	0.554	0.554	0.539	5.676	5.661	1.003
la12	0.367	0.426	0.381	291.367	291.314	1.000
new3	0.535	0.535	0.487	984.767	979.095	1.006
sports	0.242	0.238	0.106	692.814	679.757	1.019
tr11	0.645	0.645	0.633	0.180	0.176	1.021
tr12	0.472	0.553	0.523	0.096	0.094	1.015
tr23	0.252	0.260	0.223	0.045	0.044	1.019
tr45	0.363	0.555	0.495	0.663	0.649	1.022

対する階層クラスタリングにおいて MVS を用いた場合でも，cosine 類似度を用いた従来手法と同等の時間で実行可能であることがわかる．

3.7.6　$d_i^{\mathrm{T}} D$ の事前計算による貢献に関する評価

3.4.1 節では，$d_i^{\mathrm{T}} D$ を事前に計算することで，内積の計算回数を定数倍削減する手法を提案した．また，データ集合の次元数 m が $m > n$ であるときに cosine 類似度行列の事前計算を用いて更に高速化をはかる手法を提案した．これらの手法が階層クラスタリングの高速化に与える影響を実験的に評価する．この実験では，

手法 1 内積を 3 回計算する手法 ($O(3mn^2)$)

手法 2　D 及び $d_i^{\mathrm{T}}D$ の事前計算を用いる手法 ($O(mn^2 + mn + mn)$)

手法 3　cosine 類似度行列の事前計算を用いる手法 ($O(mn^2 + n^2 + n^2)$)

を用いた場合についてそれぞれ類似度行列の初期化をおこなったときの処理時間を 3 回計測しその平均を比較する．表 3–3 にその結果を示す．

表 3–3: 類似度行列初期化の処理時間

データ	処理時間 (sec)			データ	処理時間 (sec)		
	手法 1	手法 2	手法 3		手法 1	手法 2	手法 3
fbis	21.177	3.287	3.329	reviews	387.233	13.062	12.912
hitech	72.607	3.094	3.025	wap	22.637	1.213	1.180
k1a	77.766	2.811	2.750	la12	846.207	24.001	24.167
k1b	57.183	2.825	2.753	new3	3214.237	83.126	81.586
la1	69.185	5.584	5.511	sports	1328.157	41.948	42.865
la2	152.350	5.037	5.01	tr11	1.531	0.141	0.129
re0	7.42	0.426	0.419	tr12	0.807	0.080	0.072
re1	11.333	0.582	0.576	tr23	0.376	0.041	0.036
tr31	10.600	0.721	0.688	tr45	5.132	0.418	0.397

手法 1 では，内積の回数が他に比べて 3 倍必要であるため，処理時間もおよそ 3 倍になることが予想された．しかし実験結果では，最大で 40 倍程度まで増加する結果もみられた．この結果は式 (3.5) のうち，d_i, d_j に対して第 2,3 項で扱う総和ベクトル D の疎性が著しく低くなるために生じたと考えられる．本実験においては，内積対してにベクトルの疎性を利用した高速な計算手法を導入している．そのため計算をおこなうデータがそれぞれ疎性の高い単一のデータである場合には，内積の計算は高速に行えた．一方で総和ベクトル D はほぼ全ての次元に対して重みを持つ疎性の極めて低いベクトルである．この密なベクトル D に対しては，内積の高速な計算手法は大きな効果を得られない．よってデータセットの疎性が高い場合，式 (3.5) の第 1 項と第 2,3 項では，実際の処理において計算回数が大きく異なる．手法 2,3 ではこの計算回数の多い第 2,3 項について事前計算による高速化を行っているため実験上 3 倍よりも大きい高速化の効果が得られたと考えられる．

また，手法 2,3 の比較においては fbis 以外において手法 3 のほうが僅かに高速で

あることがわかった．fbis については表 3–1 にあるとおりデータ数 n よりも次元数 m が小さいためこのような結果になったと考えられる．

3.7.7　$D_a^{\mathrm{T}} D_b$ の事前計算による貢献に関する評価

3.4.2 節では，クラスタ a, b のマージ後のクラスタ間類似度を計算する上で，$D_a^{\mathrm{T}} D_b$ を事前計算することで，類似度行列の更新を高速化する手法を提案した．また，データ集合の次元数 m が $m > n$ であるときに cosine 類似度行列の事前計算を用いて更に高速化をはかる手法を提案した．これらの手法が階層クラスタリングの高速化に与える影響を実験的に評価する．この実験では，

手法 1 更新式中に $D_a^{\mathrm{T}} D_b$ を逐次計算する手法 ($O(mn)$)

手法 2 マージの発生毎に $D_a^{\mathrm{T}} D_b$ の事前計算おこなう手法 ($O(m+n)$)

手法 3 cosine 類似度行列の事前計算を用いる手法 ($O(n+n)$)

を用いた場合についてそれぞれ階層クラスタリングをおこなったときの処理時間を 3 回計測しその平均を比較する．

表 3–4: クラスタのマージの処理時間

データ	処理時間 (sec)			データ	処理時間 (sec)		
	手法 1	手法 2	手法 3		手法 1	手法 2	手法 3
fbis	18.291	16.559	17.793	reviews	80.463	71.642	72.758
hitech	15.840	13.978	13.896	wap	5.042	4.490	4.451
k1a	16.102	14.750	14.870	la12	262.018	249.488	264.416
k1b	16.042	14.680	14.787	new3	941.493	893.266	911.020
la1	40.903	37.159	37.244	sports	643.504	636.743	648.616
la2	36.846	33.355	31.949	tr11	0.176	0.052	0.047
re0	4.147	3.933	3.966	tr12	0.079	0.026	0.022
re1	5.734	5.453	5.539	tr23	0.067	0.011	0.008
tr31	1.363	0.784	0.791	tr45	0.628	0.266	0.257

表 3–4 にその結果を示す．この結果より，手法 2 は手法 1 よりも常に短い時間でのクラスタリングを可能にしている．また，理論上 $m > n$ である場合に手法 2 よ

りも高速である手法 3 の処理時間が，手法 2 に劣っている結果が多く見られた．これは，類似度行列の初期化と同様にベクトル計算を高速におこなう手法の導入による影響であることが考えられる．

3.7.8　定数 λ によるクラスタサイズの均衡化に関する評価

3.6 節では，類似度行列の初期化において定数 λ の減算によるクラスタサイズの均衡性の制御について述べた．ここでは λ の導入によりクラスタサイズの均衡化が図れていることを実験により示す．この実験では，3.7.4 と同様の $\lambda = \frac{2}{n-2}$ を用いた BMVS と，定数項を用いない MVS でクラスタリング結果の均衡性を比較する．また，クラスタサイズの均衡性は式 (3.22) に示す fairness index を用いる．

$$\mathrm{FI} = \frac{n^2}{K \sum_{i=1}^{K} n_i^2} \tag{3.22}$$

fairness index は全てのクラスタのサイズが均等に $\frac{n}{K}$ であるとき最大の値 1 となるような，均衡性の高さを測る指標である．表 4–1 に結果を示す．この結果では，reviews を除く全てのデータセットに対して fairness index は上昇または変化なしであった．このことから，定数 λ を導入することでクラスタサイズの均衡化が可能であることがわかる．

表 3–5: クラスタの均衡性の評価

データ	fairness index		データ	fairnessindex	
	MVS	BMVS		MVS	BMVS
fbis	0.299	0.322	reviews	0.385	0.382
hitech	0.483	0.492	wap	0.291	0.291
k1a	0.254	0.254	la12	0.272	0.373
k1b	0.343	0.343	new3	0.314	0.314
la1	0.373	0.377	sports	0.189	0.191
la2	0.285	0.384	tr11	0.507	0.507
re0	0.453	0.453	tr12	0.232	0.297
re1	0.258	0.258	tr23	0.465	0.472
tr31	0.561	0.561	tr45	0.156	0.291

第4章

ユークリッド距離に基づいた多観点距離

Nguyen らの研究 [2] では，基盤となる類似度を cosine 類似度として MVS を開発していた．本章では，cosine 類似度以外において，MVS と同様のアイデアに基づいた類似度の開発を試みる．一般に広く利用される類似度としてユークリッド距離がある．そこでユークリッド距離に対して MVS のように複数の基準点から距離を計算した結果を総合的に評価することで，より強固な多観点による距離を定義する．また，その距離をユークリッド距離を用いた非階層クラスタリングの代表的手法である k-means に適用したクラスタリング手法を提案する．さらに，実験により k-means を用いたクラスタリング結果を改善することを示すことでその有用性を示す．

4.1 ユークリッド距離に基づく多観点距離の定義

Cosine 類似度は式 (2.5) に示すように，2 点のなす角度の大きさによって類似性を評価する．角度はこの 2 点と基準点の 3 つによって定まる．基準点は通常の cosine 類似度では原点であり，MVS はこの基準点を個々の入力データへ移動させることで類似度に影響を与えていた．

しかしながら，ユークリッド距離では基準点を移動させても距離は不変である．実際に原点を 0 から任意の基準点 v に移動させた時，2 点 x, y のユークリッド距離

は以下の式で得られる．

$$\begin{aligned} dist(x-v, y-v) &= \| (x-v) - (y-v) \| \\ &= \| x-y \| \end{aligned}$$

しかし，これは単なる x, y 間の距離 $dist(x, y)$ と同様であることがわかる．これは，ユークリッド距離 $dist(x, y)$ が基準点に依存せず純粋に x と y のみによって定義されているためである．そのため，多観点に基づくユークリッド距離を定義するためには，cosine 類似度に対する MVS とは異なる手段によって基準点の影響を与える必要がある．

ユークリッド距離は基準点に依存しないが，絵画や作図の世界では基準点 (視点) がユークリッド距離に影響を与える遠近法が広く利用されている．遠近法では視点から遠くにある 2 点間の距離を小さく，また視点から近い 2 点の距離を大きく見せる．

本研究では，この遠近法のアイデアをベースとして，2 点 x, y 間の距離を基準点 v に基づいて変化させるような距離を提案する．基準点 v による遠近法の効果を受けた距離 $dist_v(x, y)$ を式 (4.1) のように定義する．

$$dist_v(x, y) = w_{x,y}^{(v)} dist(x, y) \tag{4.1}$$

ここで，$w_{x,y}^{(v)}$ は基準点 v に基づく x, y への遠近法の効果による重みであり，x, y と基準点 v との距離が大きいほど小さな値を取るように定める．例えば $w_{x,y}^{(v)}$ を式 (4.2) のように x, y の中点と v の距離の逆数にすることが考えられる．

$$w_{x,y}^{(v)} = \frac{1}{dist(\frac{x+y}{2}, v)} \tag{4.2}$$

しかし式 (4.2) の定義では，x, y と v が極めて近い場合に $dist_v(x, y)$ が大きくなりすぎる恐れがある．特に $dist(\frac{x+y}{2}, v) = 0$ である場合，$w_{x,y}^{(v)}$ は無限大になる．

この問題を解決するために式 (4.3) のように $w_{x,y}^{(v)}$ の定義に値が 1 より大きくならないような制限を加える．

$$w_{x,y}^{(v)} = \min\left(\frac{1}{dist(\frac{x+y}{2}, v)}, 1\right) \tag{4.3}$$

この条件により $w_{x,y}^{(v)} = (0, 1]$ となるため $dist_v(x, y)$ は $dist(x, y)$ よりも大きくならないことが保証される．ただし，$dist(\frac{x+y}{2}, v) > 1$ であるとき $dist_v(x, y) = dist(x, y)$ であり基準点の影響を受けなくなり，またそのようなデータ組 (x, y, v) の比率は入

力データ集合に依存する．そこで，$w_{x,y}^{(v)}$ の定義を制御係数 $\alpha = (0, 1]$ を加えた式 (4.4) として再定義する．

$$w_{x,y}^{(v)} = \min\left(\frac{\alpha}{dist(\frac{x+y}{2}, v)}, 1\right) \tag{4.4}$$

図 4–1 に定数 α を用いた制御の有無による $w_{x,y}^{(v)}$ の取りうる値をそれぞれ示す．α を用いることで $dist(\frac{x+y}{2}, v)$ の値が定数 α よりも大きい場合には遠近法的に距離を縮小する．α を小さく設定すれば $w_{x,y}^{(v)}$ は 1 以下の値を取りやすくなるため，基準点の影響をより多くのデータ対 (x, y) に与えることができる．

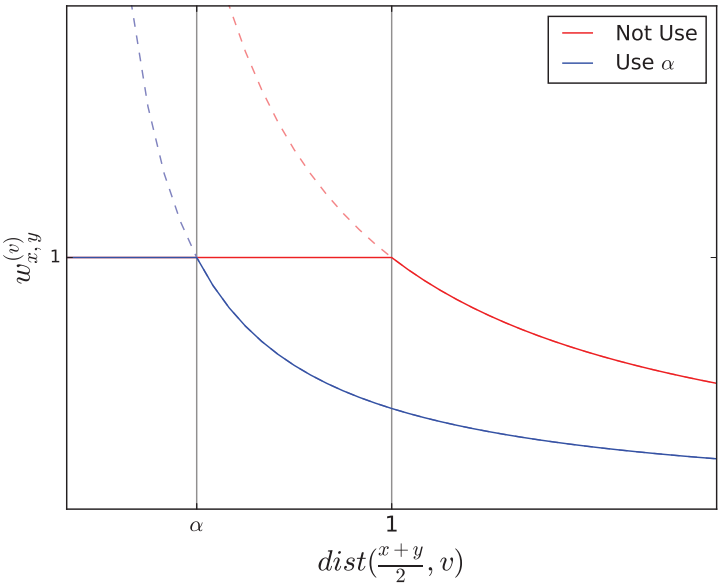

図 4–1: 定数 α による $w_{x,y}^{(v)}$ の制御

MVS ではデータ集合の複数の点を基準点として，各基準点に基づく cosine 類似度の平均から類似度を計算していた．同様に n_V 個の基準点からなる基準点集合 V を入力データ集合から作成し，各基準点 v による $dist_v$ の平均を得ることで分布に適応的な多観点距離 (Multiviewpoint-Based Distance:MVD) を式 (4.5) のように定義する．

$$\mathrm{MVD}(x, y \mid V, \alpha) = \frac{1}{n_V} \sum_{v \in V} dist_v(x, y) \tag{4.5}$$

この式に示す MVD では距離計算が n_V 回繰り返されることから，その計算量は通常のユークリッド距離の n_V 倍になることがわかる．

4.2　k-means への多観点距離の導入

本節では，前述の MVD を非階層クラスタリングの代表的手法である k-means へ導入した手法 MVD k-means を提案する．k-means では，各データを最も近い重心を持つクラスタに所属させる手続きに繰り返しにより最適なクラスタリング結果を得る．そのため，MVD を用いた場合でも n 個のデータ $d_i(i = 1, \ldots, n)$ と K 個のクラスタ重心 $\mu_k(k = 1, \ldots, K)$ との間で距離の計算を行い，d_i を最も近い重心を持つクラスタに所属させる．

MVD k-means では，MVD(d_i, μ_k) における基準点集合 V_k を全クラスタの重心集合 $M = \{\mu_1, \ldots, \mu_K\}$ から μ_k を除いた $K - 1$ 個の点と定義する．よって，データの次元数が m であるならば MVD k-means において 1 回の距離計算に必要な計算量は $O(mK)$ である．

また，MVD k-means での目的関数は式 (4.6) で表される．

$$\begin{aligned} Obj(S) &= \sum_{k=1}^{K}\sum_{i=1}^{n} r_{ik} \left\{\mathrm{MVD}(d_i, \mu_k \mid V_k, \alpha)\right\}^2 \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n} r_{ik} \left\{ \frac{1}{K-1} \sum_{v \in M \setminus \mu_k} dist_v(d_i, \mu_k) \right\}^2 \end{aligned} \tag{4.6}$$

アルゴリズム 5 に MVD k-means の流れを示す．MVD k-means では，次の 2 ステップの反復により最適なクラスタ分割をおこなう．

1. 現在のクラスタ構成に基づく各クラスタ重心の計算

2. 各データの最も近い重心をもつクラスタへの所属

ここで ϵ は収束条件を定める閾値であり，目的関数の 1 ステップ前との差分 ΔObj がこの値を下回った時クラスタリングを終了する．また，T はクラスタリングにおける反復処理の回数上限であり，反復回数が T 回に達した時クラスタリングを終了する．

アルゴリズム 5 MVD k-means

Input: $S = \{d_1, \ldots, d_n\}, r = \{r_{ik} \mid i = 1, \ldots, n\ k = 1, \ldots, K\}, \alpha, \epsilon, T$

1: $t = 1$
2: **while** $\Delta Obj > \epsilon$ **and** $t \leq T$ **do**
3:　**for** $k = 1, \ldots, K$ **do**
4:　　$\mu_k = \frac{1}{n_k} \sum_{d_i \in S_k} d_i$
5:　**for** $i \leftarrow 1, \ldots, n$ **do**
6:　　**for** $k \leftarrow 1, \ldots, K$ **do**
7:　　MVD$(d_i, \mu_k \mid M, \alpha)$ の計算
8:　　$r_{ik} = \begin{cases} 1 & (k = \arg\min_k \mathrm{MVD}(d_i, \mu_k)^2) \\ 0 & (\text{otherwise}) \end{cases}$
9:　$\Delta Obj = Obj_t - Obj_{t-1}$

Output: クラスタ所属 r

4.3　評価実験

前節まででは，ユークリッド距離に MVS のような複数の基準点に基づく距離定義を導入した MVD を提案した．また，ユークリッド距離に基づいた非階層クラスタリング手法である k-means について MVD を導入した MVD k-means を提案した．

そこで本節では以下の流れによる実験をおこなうことで，MVD k-means の有用性を示す．

1. k-means を用いたクラスタリングをおこない，そのクラスタ分割結果を得る．
2. k-means の結果をクラスタ分割の初期値として与えて MVD k-means により再クラスタリングをおこなう．
3. k-means 及び MVD k-means のクラスタ分割を入力データの真のラベルと評価し，その比較により MVD k-means が k-means のクラスタリング結果を改善できるかを評価する．

また，処理時間については k-means よりも大きくなることが理論的に明らかであるため評価をおこなわない．

4.3.1　実験準備

この実験において用いるデータセットは 3.7.1 節の実験に用いた表 3–1 のうち new3 を除いた 17 個を用いる．new3 の除外は，データセットのサイズ及び次元数が大きく実験が困難であったためである．また，これらに対する前処理は 3.7.1 節と同様の行程をおこなう．実験環境については 3.7.3 節と同様のものを用いる．k-means の初期化には k-means++[8] を用いる．また，そのクラスタリング結果を MVD k-means のクラスタラベル初期値として用いる．k-means と MVD k-means における反復回数の上限は 100 回とする．また，目的関数の収束条件は $\epsilon = 0$ とする．よって目的関数が完全に収束するか，反復が 100 回に至ることでクラスタリングが終了する．MVD には重み制御のパラメタとして α を用いていた．この実験において $\alpha = 0.1$ としてクラスタリングをおこなう．

4.3.2　分類精度評価

表 4–1 に，MVD k-means と k-means のクラスタリング精度の評価をおこなった結果を示す．この実験では 17 個のデータセットのうち 16 個において MVD k-means を用いることで k-means のクラスタリング結果を改善する結果となった．また，NMI の分散は 14 個のデータセットの実験結果において k-means よりも小さくなっているため，MVD の導入はクラスタリング結果の安定性の向上に寄与できることが考えられる．この実験中で MVD k-means において反復回数の上限によるクラスタリングの終了は発生しなかった．

表 4–1: 分類精度の評価

データ	分類精度 (分散)	
	MVD k-means	k-means
fbis	0.586(6.17E-05)	0.579(2.28E-04)
hitech	0.309(1.06E-04)	0.268(2.85E-04)
k1a	0.570(2.60E-04)	0.548(2.25E-04)
k1b	0.589(1.18E-03)	0.608(1.21E-03)
la1	0.537(1.76E-03)	0.447(4.83E-03)
la2	0.511(6.25E-04)	0.399(1.16E-03)
re0	0.454(2.49E-04)	0.440(2.91E-04)
re1	0.538(3.05E-04)	0.513(5.88E-04)
tr31	0.498(4.52E-03)	0.466(5.05E-03)
reviews	0.451(1.02E-02)	0.397(7.53E-03)
wap	0.570(1.42E-04)	0.553(5.65E-04)
la12	0.578(7.20E-04)	0.452(2.64E-03)
sports	0.639(6.40E-05)	0.440(1.14E-04)
tr11	0.657(1.13E-03)	0.627(1.83E-03)
tr12	0.635(7.07E-04)	0.595(1.55E-03)
tr23	0.352(8.54E-04)	0.342(1.25E-03)
tr45	0.674(2.79E-03)	0.663(2.59E-03)

第5章

結論

本研究では，Nguyen らが提唱した MVS に関して，2 つのテーマを取り扱った．

1 つ目は，多観点な cosine 類似度を階層クラスタリングについて適用した手法の開発である．Cosine 類似度に関する Nguyen らの MVS は，非階層クラスタリングにのみ用いられていた．そこで，本研究ではこの類似度を凝集型階層クラスタリングに適用した．MVS の単純な導入によって階層クラスタリングの計算量の増加が生じるため，クラスタ間類似度行列の初期化及びマージ後のクラスタ間類似度の更新を高速化する手法を開発した．これにより，MVS を適用した場合においても一般的な階層クラスタリングと同様に計算量 $O(mn^2 + n^2 \log n)$ を実現した．さらに文書データを用いた実験により，MVS を用いた階層クラスタリングが既存手法と同程度の計算時間で，より高い分類精度を示すことを確認した．

2 つ目は，cosine 類似度以外の類似度に対する多観点類似度の考案である．本研究では一般に類似度指標として広く用いられているユークリッド距離を基盤とした多観点距離 (MVD) を提案した．ユークリッド距離では MVS のように基準点との差分ベクトルを用いた影響は与えることができない．そのため，絵画や作図などで用いられる遠近法のように基準点からの距離に応じて元々の距離の拡縮を行うことで，ユークリッド距離に対する基準点の影響を定義した．また，MVD を非階層クラスタリングの代表的手法である k-means に適用した MVD k-means を開発した．さらに文書データを用いた実験により，k-means によるクラスタリング結果を MVD k-means により改善できることを示した．

本研究の今後の課題を述べる．MVD では，遠近法の効果による基準点 v から 2 点 x, y に対する重み $w_{x,y}^{(v)}$ を制御するために定数 α を導入していた．この α は現段階では人為的に与えるパラメタであり，さらに最適な α は入力データ集合ごとに

異なる．入力データ集合から最適な α を算出する手法を構築することで人為的な操作による結果の変動を防ぐとこができると考えられる．

本研究で開発した MVD k-means の計算量は $O(nmK^2)$ であり，通常の k-means の $O(nmK)$ よりも計算量が大きい．この計算量を小さくする手法を構築することで MVD k-means の有用性をさらに高められる．

また，MVD k-means では，実験において目的関数の振動は確認されなかった．この結果から MVD k-means の目的関数について収束性があることが期待できる．MVD k-means についてより詳細に解析をおこなうことで理論的に目的関数の収束性を証明できる可能性がある．

参考文献

[1] I.S. Dhillon and D.S. Modha, “Concept decompositions for large sparse text data using clustering,” Mach. Learn., vol.42, no.1-2, pp.143–175, 2001.

[2] D.T. Nguyen, L. Chen, and C.K. Chan, “Clustering with multiviewpoint-based similarity measure,” IEEE transactions on knowledge and data engineering, vol.24, no.6, pp.988–1001, 2012.

[3] J.B. MacQueen, “Some methods for classification and analysis of multivariate observations,” Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol.1, pp.281–297, University of California Press, 1967.

[4] G. Salton and M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., New York, NY, USA, 1986.

[5] S. Jayaprada, A. Aswani, and G. Gayathri, “Hierarchical divisive clustering with multi view-point based similarity measure,” Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, pp.483–491, 2014.

[6] S. Bickel and T. Scheffer, “Multi-view clustering.,” Proc. ICDM 2004, pp.19–26, 2004.

[7] G. Karypis, “Cluto-a clustering toolkit,” Technical report, MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE, 2002.

[8] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp.1027–1035, SODA ’07, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.

謝辞

本研究をおこなうにあたって多数のご指導とご助言をいただいた古賀久志准教授，南泰浩教授，に心から感謝いたします．日頃から本研究に関して活発なご意見，ご助言をいただいた戸田貴久助教授と中鹿亘助教授に深く感謝いたします．多忙の中，多くのご助言，ご協力をいただいた柳生智彦客員准教授と鈴木一哉客員准教授に深く感謝いたします．また，研究室での生活や研究の様々な場面でアドバイスをいただきました南・古賀・戸田・中鹿研究室の学生の皆さま，すでにご卒業された先輩方に心から感謝いたします．

平成 30 年 1 月 29 日

図一覧

表一覧