「文書クラスタリングの技法」ゼミ

①：SKWIC法
②：Scatter/Gather のアルゴリズム
③：C^3M および C^2ICM アルゴリズム

坂元　唯

①SKWIC法

「意味と定義」

◆k-meansを拡張し、最適なクラスタリング＋最適な語の重みを同時に計算するアルゴリズム。

定義

・文書d_iとクラスタC_kの重心ベクトルの比類似度を以下の式で定義。

→この値が小さいほど、そのクラスタは類似していることになる。

$$\hat{s}(d_i, C_k) = \sum_{j=1}^{M} \tilde{w}_{kj} \hat{w}_{ijk} \qquad \cdots ①$$

- ここで上記の式より、

$$\hat{w}_{ijk} = \frac{1}{M} - x_{ij} m_{kj} \qquad \cdots ②$$

であり、この式中のm_{kj}は、重心ベクトルの(0)式におけるj番目の要素。

- SKWIC法では、これらの量を使って、次のような目的関数を定義し、これを最小とするようなm_{kj}と$\hat{w}_{ijk}$とを算出する。つまり、

$$J_s = \sum_{k=1}^{L} \sum_{i:d_i \in C_k} \sum_{j=1}^{M} \tilde{w}_{kj} \hat{w}_{ijk} + \sum_{k=1}^{L} \left(\delta_k \sum_{j=1}^{M} \hat{w}_{kj}^2 \right) \qquad \cdots ③$$

である。ただし、

$\tilde{w}_{kj} \in [0,1]$　かつ、$\sum_{j=1}^{M} \tilde{w}_{kj} = 1, k = 1, \ldots, L$　を条件とする。

- この問題をLaglangeの未定係数法を使って解くと、

$$\tilde{w}_{kj} = \frac{1}{M} + \frac{1}{2\delta_k} \times \sum_{i:d \in C_k} \left[\left(\frac{1}{M} \sum_{j=1}^{M} \hat{w}ijk \right) - \hat{w}_{ijk} \right] \cdots ④$$

を得る。

- なお δ_k は、目的関数③式における第１項と第２項とのバランスを調整するためのパラメータ。実際には、クラスタリングの過程の中で反復的に値が算出される。

「SKWIC法の手順」

ⅰ. クラスタの個数Lを決め、L件の文書を無作為抽出して、それらを各クラスタの重心とする。　→初期化

ⅱ. $\hat{w}_{ijk}$ を式②に従って計算する。

ⅲ. 式④を使って $\hat{w}_{jk}$ を更新する。

ⅳ. 式①を使って、$\hat{s}(d_i C_k)$ を計算する。

ⅴ. $\hat{s}(d_i, C_k)$ に基づいて各文書$d_i,\ldots,d_N$を最も近いクラスタに分類する。

ⅵ. 新たに求められたクラスタに対して、重心ベクトル（0）式を計算する。
ただし、$\tilde{w}_{kj} = 0$ なら $m_{kj} = 0$ とする。

ⅶ. もし再計算された重心ベクトルが前段階と同じなら、処理を終了。そうでなければ再計算して、手順 ⅱ へ戻る。

「SKWIC法のまとめ」

- この手法はクラスタの個数Lを先験的に決め、そのうち、その重心に最も近いクラスタに文書を割り当てる事。

  →基本的にはk-means法。

- 特徴（すなわち語）の重みを同時に最適化することに独創性がある。

- 重みを使えば、各クラスタを特徴づける為の「最適な」語の集合の特定が可能。

  →文書クラスタリングにとって興味深い方法。

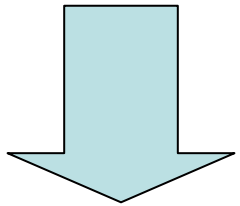
② Scatter/Gather のアルゴリズム

「意味」

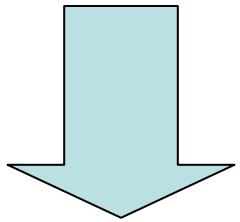
- Scatter/Gatherは、クラスタリングを活用することによって大規模な文書集合の通覧・検索を可能にするシステム。
- 「Scatter」とは、1つまたは複数のトピックに関する文書集合をクラスタリングして、更に詳細に分割する事。
- 「Gather」は、その分割された集合から、関心のあるものをいくつか拾うこと。

「仕組み」

「Scatter」と「Gather」を交互に繰り返す



大規模な文書集合



小規模な文書集

（利用者の関心に密接な関連を持つ）

「処理手順①」

- クラスタリングの基本的な方法はk-means法に基づく単一パス・アルゴリズム。

- 特徴
  ・・・L個のクラスタベクトルを設定するために、処理時間がより必要となる階層的クラスタ分析を使う。
  →Buck-shot、Fractionaionという2つのアルゴリズム。

「処理手順②」

ⅰ．L個の中心点（クラスタベクトル）を見つける。

ⅱ．各文書を1つの中心点に割り当てる。

→k-means法

ⅲ．分割の精緻化（refinement）を試みる。

- 標準的な階層的クラスタ分析が適用されるのは、このうちの第１段階である。
- なお、ここでは、階層的クラスタの分析の計算量は、n件の文書を処理する場合に$O(n^2)$であると考えておく。

「Buckshot」

- 仕組み
  - 文書集合全体から$\sqrt{LN}$ 件の文書を無作為に抽出して、それに対して、階層的クラスタ分析を適用する。

- 特徴
  - 結果は抽出に依存。
  - 処理が速い。

- 結論
  - Scatter/Gatherにおける階層的クラスタリングの目的は、あくまでクラスタベクトルの設定なので、無作為抽出された標本が母集団の様子を正しく反映していれば十分。

「Fractionation」

- 仕組み
  - ・最初に文書集合全体を、それぞれm件の文書からなる小さな集合に機械的に分割。
  - ・それぞれに対して階層的クラスタ分析を適用する。
  - ・得られた結果を書くクラスタの単一文書とみなして、同一の手順を繰り返す。

- 特徴
  - ・Buckshotよりも処理時間を要する。

- 結論
  - ・一応全ての文書を対象にして分析するので、より良い結果が期待出来る。

「アルゴリズム」

ⅰ. 文書集合全体DをN/m個の排他的部分集合に機械的に分割。

ⅱ. それぞれに対して、pm個のクラスタが得られるように、階層的クラスタ分析を適用する。

→結果、全部でN/m×pm=pN個のクラスタを得る。

ⅲ. ・$C_{k,h}$をk番目の部分集合におけるh番目のクラスタとする。
・これらの$C_{k,h}$を単一の文書とみなし、合計pN件の文書に対して同様の手順を再び適用する。

→ p^2N 個のクラスタが得られる。

ⅳ. この処理をr回繰り返して、$p^r N \leq L$ となった時点で手順を終了し、最後に残ったクラスタのベクトルを採用する。

「Scatter/Gatherのアルゴリズム①」

- 定義

・文書ベクトルの要素・・・・・ $w_{ij} = \sqrt{x_{ij}}$

・クラスタCkのベクトル・・・・

$$C_k = \sum_{i:d_i \in C_k} d_i / \|d_i\| \quad \Rightarrow \quad \hat{w}_{kj} = \sum_{i:d_i \in C_k} \frac{w_{ij}}{\sqrt{\sum_{j=1}^{M} w_{ij}^2}}$$

・これがScatter/Gatherの具体的な「中心点」となる。

「Scatter/Gatherのアルゴリズム②」

ⅰ．L個の中心点が決まれば、L件の文書を順に各中心点に割り当てていく。
　　・・・L個のクラスタのベクトルとの類似度を計算し、その値の最も大きなクラスタに
　　その文書を配分する作業。

ⅱ．最後に、その結果として得られたL個のクラスタを精緻化する。
　この作業は（分化）と（結合）から成る。

「Scatter/Gatherのアルゴリズム③」～分化と結合～

- 分化（split）
  ・その時点で得られている各クラスタに対して、Buckshotを適用する。

- 結合（join）
  ・2つのクラスタにおける「主用語」が共通している場合に、両者を併合する操作。

※「主用語」とは、クラスタのベクトル中の重みによって語を並べた時の上位y個を指す。
・・・具体的には、2つのクラスタ間で共通する「主用語」の数を調べ、その共通語数がある閾値を超えていれば、両者を併合する。

「Scatter/Gatherのアルゴリズム　～まとめ～」

- 用途

  ・実際に、利用者に対して高速で応対しなければならない場合等・・・

  → Buckshot

  ・利用者の問い合わせ以前にクラスタリングする場合等・・・

  → Fractionation

- 階層的クラスタの分析方法

  ・凝集型の郡平均法が利用されている。

  ・文書集合が前もって階層的に構造化されていれば、Scatter/Gatherシステムを使う際に、階層的クラスタ分析を実行する必要は無く、その分、応答速度が速くなる。

  ・ただし、事前に構成された階層構造が必ずしも、情報要求に合うとは限らず、不要な文書も結果に含まれてしまう可能性がある。

  → 任意の文書集合に対して、既存の階層構造から必要なクラスタを抽出し、それに基づいて、クラスタリングを高速で実行するアルゴリズム

③C^3M および C^2ICM アルゴリズム

「解説」

- k-means法では、通常、クラスタの重心ベクトルとの類似度に基づいてクラスタリングが実行される。

- k-medoid法では、クラスタを1つの分類対象によって代表させ、それとの類似度によって、対象を分類する。

「C^3Mの具体的な手順」
～C^3Mに基づくアルゴリズム～

ⅰ. “cluster seed power”に基づいてクラスタの種子点をL個選び、種子点の集合Dsを作り、i=0と設定する。

ⅱ. i←i+1とし、文書d_iを読む。

ⅲ. もし、N件の文書を全て読み終わっていたならば、クラスタリングを終了する。そうでなければ、ⅳに進む。

ⅳ. もし、d_i種子点ならば、ⅱに戻る。そうでなければⅴに進む。

ⅴ. d_iを最も覆っている種子点のクラスタにd_iを割り当てる。その後、ⅱに戻る。

「cluster seed power」

- 文書diに対して、

$$p_i = \delta_{ii}\left(1-\delta_{ii}\right)\sum_{j=1}^{M} b_{ij}$$

で定義される。

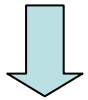
- 基本的には、「独自性」が中程度の文書の値が大きくなるように、$\delta_{ii}\left(1-\delta_{ii}\right)$ を使い、更に文書に含まれる語数 $\sum_{j=1}^{M} b_{ij}$ を乗じている。

- 特徴
  - 前述の v の段階で、特にクラスタを1つに絞らなければ、重複を許すクラスタリングとなる。
  - どの種子点にも割り当てられない文書は除外しておき、最後にそれらをまとめて「その他」クラスタとする。

「C^2ICM」

- 意味
  - ・C^3Mによって構成されたクラスタ集合に対して、新たな文書の追加・削除を実行するためのシステム。

- その他
  - ・F^2ICM・・・C^2ICMアルゴリズムを拡張し、文書の価値が時間的に衰退していくことを組み込んだ手法。



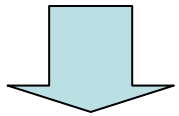
  - ・古い文書よりも新しい文書の法がクラスタリングの結果に貢献するように、各文書に対して、指数関数的に減少する重みを設定し、これを種々の計算に活用する。

「Leader-followerアルゴリズムの応用」

- 手順

ⅰ．クラスタ個数の代わりに、文書クラスタとの類似度の閾値を前もって設定しておく。

ⅱ．分類対象を順に処理していく中で、この閾値に基づいて自動的にクラスタが形成されていく。



・単純なk-means法では、クラスタ個数及びそれらのベクトルを先験的に与えなければならないが、それを回避できる単一パスとなる。