階層的な規則を用いた文書のクラスタリング

佐々木 稔　北 研二
徳島大学工学部

1 はじめに

近年，インターネットの普及とともに，WWWを代表とするネットワーク上の大量の電子データを個人が取り扱えるようになり，テキストデータの山のなかで必要な情報を取り出す機会が増加している．このような状況を反映して，一般の多様な文書を対象とした文書のクラスタリングなどの重要性が現実問題として企業や個人のレベルにまで認識されている．たとえば，Yahooなどを代表とする組織的なインデックスサイトや個人レベルでのリンク集などさまざまなクラスタリング情報，コメント情報などを手作業で作成しているサイトがあり，インターネット内のテキストデータ中の情報を見つける際の有力な手段であると実感することが多い．

しかし，これらのテキストの分類，索引付けや注釈付けなどの整理の作業には多大な労力と時間がかかるため，これまでに自動的にクラスタリングを行う研究が数多く行われている．本稿では，Cohenの提案した規則学習を行うアルゴリズムRIPPER[1][2]の問題点を指摘し，これを改善するために複数の分野を共通して満たす規則を生成するアルゴリズムを提案する．

2 規則に基づく文書クラスタリング

2.1 RIPPER

規則に基づくクラスタリングアルゴリズムは学習データを満たすような規則を作成することでクラスタリングを行うもので，ノイズのある大きなデータ集合が与えられても有効である．このアルゴリズムにCohenの提案した規則学習アルゴリズムRIPPER(for Repeated Incremental Pruning to Produce Error Reduction.)がある．RIPPERは規則を作成するために処理していない文書集合をgrowing setとprunning setに分割する．このときgrowing setとprunning setに偏りがあるといけないので，全データの2/3をgrowing setに，残りの1/3をprunning setに分割する．次に規則を作成するが，まず空の条件集合から始めすべてのデータを満たすように，規則に条件を次々と付け加える．

規則を作成した後，prunning setを用いて規則から条件を削除する．規則を削除するために，ここでは処理を簡単にするために規則の一番最後，すなわち最も右側の条件から順番に削除し，簡単化を行うようにしている．このとき次の関数を最大化するように削除する条件を選択する．

$$v(Rule, PrunePos, PruneNeg) \equiv \frac{p-n}{p+n} \quad (1)$$

ここで，$PrunePos$はprunning setで現在考慮しているクラスタであるデータ数，$PruneNeg$はprunning setで現在考慮しているクラスタでないデータ数，pは$PrunePos$のうちで規則を満たす数，nは$PruneNeg$のうちで規則を満たしているものの数をそれぞれ表している．このvの値がどの条件を削除しても値が改善されなくなるまで繰り返す．また，規則を加えた後に規則集合の全記述長を計算し，規則のなかで最も短い記述長と最も長い記述長との差がdビット以上になれば終了，または考慮するクラスタが無くなった場合も終了する．

2.2 RIPPERの問題点

RIPPERが生成する規則は一次元的にクラスタが決まる規則，すなわち，特定の単語が文書内に存在していれば，それだけでクラスタが決ってしまう規則である．したがって，“サッカー”と“アメリカンフットボール”のように似通ったクラスタに属する文書に対してクラスタリングを行った場合，しっかりとした分類を行うために“ボール”などのような共通

して出現する単語については削除される．したがって，“ヘディング”などの特定の単語が規則の中に出現し，規則の数が少なくなることが予想される．

この問題を解決する試みとして，非階層的なクラスタリングアルゴリズムであるRIPPERを繰り返し実行することにより階層構造を作成する実験を行う．World Wide Web(WWW)中のインデックスサービスやリンク集のサイトから索引情報のページを取り込み，分類カテゴリとリンク先ページのURL(Uniform Resource Locater)の組を取り出す．このURLの集合をもとにして，そのリンク先ページの内容を取り出し，形態素解析にかけ，名詞と判定されたものだけを取り出す．名詞ではないものは日本語などの言語に必ず存在する機能語であると考えて，ここでは扱わない．

これにより作成された学習データとクラスタのペアを一組のデータセットとし，同一クラスタ中にあるすべてのリンク先ページについてデータセットを作成する．いくつかのクラスタのデータセットから先に述べたRIPPERを用いて規則集合を作成する．できた規則集合を用いて今度は先に規則集合を作成したデータセットのあるクラスタの下の階層に存在するデータセットがどのクラスタに存在するかを調べ，再現率を求める．

WWW中のリンク集のサイトとして有名なYahooに登録されているデータを用い，その中の美術，インターネット・コミュニケーション，音楽，小規模事業情報の4つのカテゴリに対して実験を行った．その結果，再現率は46%となり，それほど良い結果とはならなかった．

3 階層的規則に基づく文書クラスタリング

3.1 問題の有効な解決策

繰り返しRIPPERを用いて学習を行ったとき，近い内容を持つ文書はそれらの表すクラスタにクラスタリングされ，それらをまたクラスタリングすることによりさらに詳しい分類が可能になると予想される．しかし，その場合同じクラスタにある文書の集合だけを用いてクラスタリングを行うので，どうしてもデータの数は少なくなり，ノイズの影響が大きくなり正しくクラスタリングが行われない．また，規則学習が同じデータを用いて繰り返し行われるため，同じ単語が規則中に存在している可能性がある．

これらの問題を解決するために，特定のクラスタについての規則を学習により求めるだけではなく，複数の分野にだけ存在する単語についての規則を求めることで，その文書がひとつのクラスタではなく，複数のクラスタを共に満たすことができるのではないかと考える．そうすることにより先に行った実験で問題になったような，ふたつの階層に同じ単語が存在することがなくなり，見た目にも分かりやすい規則にすることができる．それには規則を階層化することによってふたつのクラスタに共通な単語を見つけだし，それをクラスタに共通的な語とみなす．その規則によって，複数の分野について書かれた文書であるとクラスタリングされた文書は，特定の事柄のみを書いたものではなく，様々な内容について書かれたものであると認識することができる．

RIPPERの生成する規則の簡単な例として，次のような規則が学習によって生成される．

$$\text{野球} \leftarrow \text{“ホームラン”} \in \textit{document}.$$

これはdocumentという文書中に“ホームラン”という単語が存在していれば，野球というクラスタに属する文書であることを表す．これを階層的に表すためにふたつのクラスタを同時に満たすクラスタを新しく作成し，新しいクラスタとそれまでに存在していたクラスタとの規則を作成する．ふたつのクラスタに共通する単語が新しいクラスタに属する規則として，規則集合に登録する．簡単な例を挙げると，次のような規則を作成することにより規則が階層的に生成される．

$$\text{スポーツ} \leftarrow \text{野球} \in \textit{clusters}, \ \text{サッカー} \in \textit{clusters}.$$

$$\text{スポーツ} \leftarrow \text{“ボール”} \in \textit{document}.$$

これは“ボール”という単語が文書中に存在していれば，新しく作られたクラスタ“スポーツ”に属する文書であることを表し，“スポーツ”というクラスタは“野球”と“サッカー”というふたつのクラスタを同時に満たしているものである．

3.2 階層的クラスタリングアルゴリズム

複数のクラスタを共に満たすことができる規則を作成し，その規則を満たす文書は複数に分野につい

て書かれた文書であるとみなすことがでるようにするために，学習によって一次元的にクラスタが決まる規則を生成するアルゴリズムであるRIPPERに変更を加える．2.2の繰り返し学習を行って分割するような方法では，学習に必要な時間が非常に大きくなるのに対して，RIPPERのアルゴリズムの中で階層的に規則を作成することができれば，ある程度の計算量は必要であるとしても，深刻な問題になるものではない．

RIPPERはすべての学習データから節の全体集合を生成した後，一度に削除を行わず，連続して節をひとつずつ削除していく．このときに削除を行うということは，その他のクラスタの中に同じ単語が存在しているために削除されるのであるから，ここで削除を行わず共通語であると考えて，これらを階層的な規則として登録を行う．RIPPERの生成する規則はひとつのクラスタに対して，それを最もよく表している単語を規則中に残している．それは，pruning setによって繰り返し単語の削除が行われても規則に残っている単語であるので，その他のクラスタにはほとんど存在しない単語であると考えることができる．このような，pruning setによって削除が行われる単語の中で，ふたつのクラスタの中でともに頻度が高い場合に，その文書はふたつのクラスタを共通に満たす文書であるとする．このアルゴリズムの概要を図1に示す．

ここで，削除された規則の中に存在する単語について，その他のクラスタとの出現頻度の比較を行う．出現頻度が低ければ，その単語はあまりそのクラスタを表しているとはいえないと考えられるので，そのような場合には次のクラスタで頻度が高いかどうかを調べる．ともにある一定の値よりもよく文書中に出現しているものが見つかったとき，ふたつのクラスタを同時に満たすクラスタを新しく定義し，新しいクラスタとそれまでに存在していたクラスタとの規則，ふたつのクラスタに共通する単語が新しいクラスタに属する規則を作成し，これらの規則を規則集合に登録する．このため，RIPPERにより学習された規則集合には全く影響を及ぼすこともなく，クラスタを代表する単語を増加させることができる．規則中の単語の数が増加することにより，テストデータを検索する際に失敗することが少なくなり，より正確な検索結果が得られると考える．こうして，規則の階層化を行い，文書が複数のクラスタを満たすものについてもクラスタリングを行うアルゴリズムを，階層的RIPPERアルゴリズムと呼ぶことにする．

```
procedure HierarchicalRIPPER(Pos, Neg)
begin
  Ruleset := 0
  while Pos ≠ 0 do
    split(Pos, Neg) into (GrowPos, GrowNeg)
      and (PrunePos, PruneNeg)
    GrowRule := Grow(GrowPos, GrowNeg)
    PruneRule := Grow(PrunePos, PruneNeg)
    if |GrowRule| − |PruneRule| > threshold
    then
      return Ruleset
    else
      add Rule to Ruleset remove examples
        covered by Rule from (Pos, Neg)
      if Removed example has higher
        score than threshold score
      then
        add Rule as a class covered by
          both classes to Ruleset
      endif
    endif
  endwhile
  return
end
```

図 1: 階層的RIPPERアルゴリズム

規則を階層化することで文書が複数の分野について書かれたものであっても，1回の処理でクラスタリングを行うことができる．すなわち，ベクトルなどでクラスタリングを行う場合は複数回の処理を行って複数のクラスタに存在するかどうかを検索しなければならない．これに対して，階層的に規則を作成することによって，複数のクラスタをひとつのクラスタとして扱うことができ，人間にも分かりやすい規則集合にもなっている．

3.3 クラスタリングの時間的効率

クラスタリングに要する時間は，規則生成後の削除の段階で削除された単語を用いているので，計算

量は階層化を行わない場合とほとんど同じである．このアルゴリズムは階層的な規則を作成する際，規則中に存在する単語を含む文書を考慮せず，それ以外の文書について共通語を抽出する．特定のクラスタに属する文書は存在しないので，階層的な規則を作成するためのデータ数が少ない．したがって，少ない計算量で階層化を行うことができる．そのために高速な計算機でクラスタリングを行う必要はなく，個人で入手した文書をクラスタリングしたいなどの場合において，規模の小さいパソコンでも処理を行うことが可能である．

4 評価実験

前節において提案した階層的RIPPERクラスタリングアルゴリズムが一般的な文書に対して有効であるかどうかを評価する．2.2節と同じ名詞のみを抽出した学習データから階層的RIPPERクラスタリングアルゴリズムにより学習を行い，学習データを満たす階層的な規則集合を作成し，テストデータでその規則集合がテストデータの表すクラスを再現できるかどうかを評価する．

4.1 実験結果

本アルゴリズムでは，階層的に規則を生成するための条件として，単語がどのくらい文書中に存在しているかを表すスコアに閾値(図1のthreshold)を設定して，規則集合から削除された単語の中でこの閾値よりも高い値を持つものを考慮する．このため，この閾値をあらかじめ決定しておく必要がある．様々な閾値について実験を行ったところ，再現率(Recall)は閾値が12の場合 $Recall = 90/198 = 45.45\%$，閾値が15の場合 $Recall = 101/198 = 51\%$となった．これは，階層化を行わない普通のRIPPERアルゴリズムで実験を行ったときの再現率である52.02%とほとんど変わらない結果となった．しかし，閾値が15の場合，2.2節で行った繰り返し処理を行う方法を用いた実験結果にくらべてよい結果となっている．

4.2 考察

実験結果として示した通り，再現率はそれほど高くなく，うまくクラスタリングされているとはいえない．これはWWWページのデータはばらつきが多く，文の数が少ないために機能語の割合が多いことが予想される．しかし，規則中に現れている単語はふたつのクラスタを代表する単語に近いものが取り出されいる．その数は閾値が大きくなるにしたがって少なくなるが，よりふたつのクラスタを共通的に代表する単語が取り出されることになる．

本アルゴリズムはひとつの単語が文書中に存在していれば，その文書の属するふたつのクラスタが決ってしまう．そのため，ふたつ以上の単語が共起することによってクラスタが決まるものを考慮していない．多くの単語が上位の規則として取り出されるが，規則の数を増やすのではなく，それを統合することでクラスタを表す規則を複数の単語で表現できると考えられる．さらに，本アルゴリズムは2種類のクラスタを同時に満たす単語を規則とし，3種類以上のクラスタを同時に満たす文書を考慮していない．この点についても改善することで，よりよい結果が得られると考えられる．

5 おわりに

本稿では，ふたつのクラスタに共通な単語を見つけだし共通語として規則を生成する階層的RIPPERアルゴリズムを提案した．これによりそのクラスタを代表する単語に近いものが取り出され，規則集合に出現する単語の数を増加させることができた．しかし，評価実験を行った結果，再現率は階層化を行う前とほとんど変化がなかった．今後，WWWからの入力データの取得方式の改良，または，規則生成方法の改良を行う必要がある．

参考文献

[1] William W. Cohen: "Fast Effective Rule Induction", Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, California, 1995.

[2] William W. Cohen: "Learning to Classify English Text with ILP Methods", In Advances in Inductive Logic Programming (Ed. L. De Raedt), IOS Press, 1995.