R による正規表現処理とテキストマイニング

石田 基広

徳島大学総合科学部

R でテキストデータを解析しよう

- 良くある手順
  - Java, C, Perl などで解析プログラム作成
  - テキストデータから必要な数値を取得
  - 抽出された数値を R に取り込み解析開始

R でテキストデータを解析しよう

- 良くある手順
  - Java, C, Perl などで解析プログラム作成
  - テキストデータから必要な数値を取得
  - 抽出された数値を R に取り込み解析開始
- R 上ですべてをシームレスに行えないのか

報告内容

- R によるテキストマイニング
  - **lsa** パッケージ

    潜在的意味インデキシング
  - **tm** パッケージ

    String Kernel によるクラスタリング

報告内容

- R によるテキストマイニング
  - **lsa** パッケージ

    潜在的意味インデキシング
  - **tm** パッケージ

    String Kernel によるクラスタリング
- R における正規表現処理
  - **gsubfn** パッケージ

テキストデータの処理

テキストマイニングに必要な古典的な処理

- 各種フォーマットやメタ情報の処理 (**XML** パッケージ)

テキストデータの処理

テキストマイニングに必要な古典的な処理

- 各種フォーマットやメタ情報の処理 (**XML** パッケージ)
- 単語の頻度を計る
- 特殊記号や stopwords を取り除く
  - "a" "about" "above" "across" "after"

テキストデータの処理

テキストマイニングに必要な古典的な処理

- 各種フォーマットやメタ情報の処理 (**XML** パッケージ)
- 単語の頻度を計る
- 特殊記号や stopwords を取り除く
  - "a" "about" "above" "across" "after"
- 大文字・小文字の統一
- stemming を行う (**Rstem,Snowball** パッケージ)
  - "Human machine interface for ABC computer applications"
  - human machin interfac abs comput application

テキストデータの処理

テキストマイニングに必要な古典的な処理

- 各種フォーマットやメタ情報の処理 (**XML** パッケージ)
- 単語の頻度を計る
- 特殊記号や stopwords を取り除く
  - "a" "about" "above" "across" "after"
- 大文字・小文字の統一
- stemming を行う (**Rstem,Snowball** パッケージ)
  - "Human machine interface for ABC computer applications"
  - human machin interfac abs comput application
- Term Document 行列を作成する

ターム・文書行列の例

term	Doc1	Doc2	Doc3	Doc4
abc	1	0	0	0
application	1	0	0	0
comput	1	1	0	0
human	1	0	0	1
interfac	1	0	1	0
machin	1	0	0	0

ターム・文書行列の例

対象とする全テキストに登場するタームを行とし，列には各文書を取る．そして各タームのある文書における頻度を要素として埋める．さらに通常は，この行列に各種の重み付けを行う．

lsa パッケージによる処理

- 潜在的意味インデキシング解析 (lsa)
- lsa_0.57 by Fridolin Wild
  - ディレクトリ内の全テキスト読込
  - オプションで stopwords を削除
  - stemming に対応 (**Rstem** による処理)
  - ターム・文書行列の生成 (重み付け)
  - 特異値分解
  - 文書どうしの類字度 (コサイン距離等) の算出
  - 新規検索タームの文書行列との類似度計算

ベンチマークによる試行例

- 九つのテクニカルメモのタイトルを利用
  - Scott C. Deerwester et al.(1990) “Indexing by Latent Semantic Analysis”
- D1 - D5 (human-computer-interaction)
- D6 - D9 (graph theory)

ベンチマークによる試行例

- 九つのテクニカルメモのタイトルを利用
  - Scott C. Deerwester et al.(1990) “Indexing by Latent Semantic Analysis”
- D1 - D5 (human-computer-interaction)
- D6 - D9 (graph theory)
  - D1: Human machine interface for ABC computer applications
  - D2: A survey of user opinion of computer system response time
  - D3: The EPS user interface management system
  - D4: System and human system engineering testing of EPS
  - D5: Relation of user perceived response time to error measurement
  - D6: The intersection graph of paths in trees
  - D7: Graph minors IV: Widths of trees and well-quasi-ordering
  - D8: The generation of random, binary, ordered trees
  - D9: Graph minors: A survey

ファイルの読み込みと行列作成

```
# 文書ディレクトリの指定
td <- ("/home/user/texts/")
# stopword をロード
data(stopwords_en)
# ディレクトリを読み込み
# ターム・文書行列作成
myMatrix <- textmatrix(td,
    stopwords = stopwords_en,
    stemming = TRUE)
```

```
# 出力を編集
> myMatrix
                docs
terms            D1 D2 D3 D4 D5 D6 D7 D8 D9
  abc             1  0  0  0  0  0  0  0  0
  application     1  0  0  0  0  0  0  0  0
  comput          1  1  0  0  0  0  0  0  0
  human           1  0  0  1  0  0  0  0  0
  interfac        1  0  1  0  0  0  0  0  0
  machin          1  0  0  0  0  0  0  0  0
  opinion         0  1  0  0  0  0  0  0  0
  respons         0  1  0  0  1  0  0  0  0
  survey          0  1  0  0  0  0  0  0  1
  syst            0  1  1  2  0  0  0  0  0
```

もとの頻度行列と検索語との距離

```
myQuery <-
    query("user interface",
    rownames(myMatrix),
    stemming = TRUE)
myMat.Que <-
    cbind(myMatrix,
    myQuery)
as.matrix(round(
    cosine(myMat.Que),
    dig = 2)[,10])
```

```
#  出力を編集
     [,1]
D1   0.29
D2   0.27
D3   0.63
D4   0.00
D5   0.27
D6   0.00
D7   0.00
D8   0.00
D9   0.00
QU   1.00
```

特異値分解

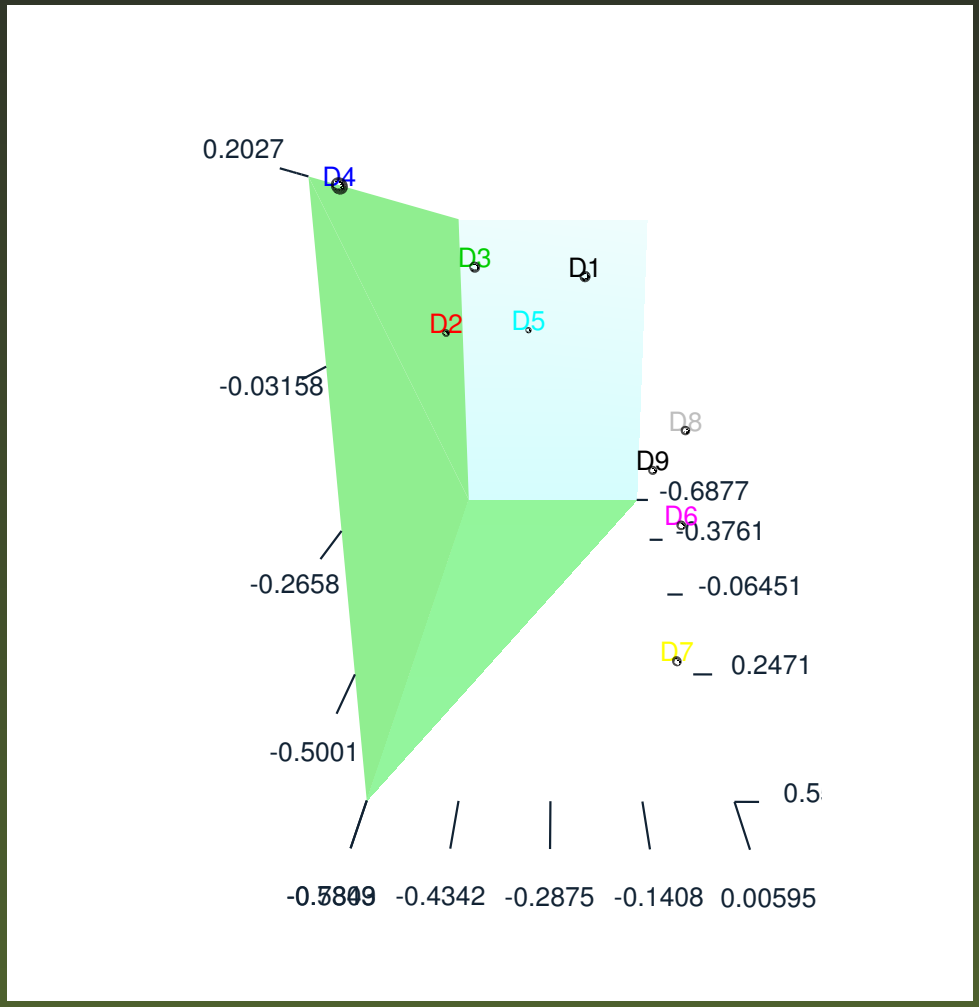
```
# LSA を実行してみる
myLSAspace <-
    lsa(myMatrix,
    dimcalc_share(0.4))
myLSAspace
round(myLSAspace$tk,
    digits= 2)
```

```
#  出力を編集
           [,1]  [,2]  [,3]
abc       -0.06  0.02  0.07
applicat  -0.06  0.02  0.07
comput    -0.22 -0.01 -0.04
human     -0.22  0.09  0.26
interfac  -0.18  0.05  0.13
machin    -0.06  0.02  0.07
opinion   -0.16 -0.03 -0.11
respons   -0.26 -0.08 -0.35
survey    -0.18 -0.16 -0.09
syst      -0.58  0.13  0.33
```

文書ベクトルの 3 次元表現

```
new3Doc <-
    t(myLSAspace$tk)
    %*% myMatrix
rgl.open()
rgl.bg(color =
    c("white", "black"))
rgl.spheres(new3Doc[1,],
    new3Doc[2,],
    new3Doc[3,])

rgl.texts(new3Doc[1,],
    new3Doc[2,],
    new3Doc[3,],
    rownames(myLSAspace$dk))
```



北 研二 他 (2002)『情報検索アルゴリズム』共立出版

3 次元文書空間での検索

```
myQuery3 <-
    query("user interface",
    rownames(myLSAspace$tk),
    stemming = TRUE )
new3Query <-
    t(myLSAspace$tk)
    %*% myQuery3
myMat.Que3 <-
    cbind(new3Doc,
    new3Query)
as.matrix(round(
    cosine(myMat.Que3),
    dig = 2)[,10])
```

```
      [,1]
D1    0.63
D2    0.98
D3    0.82
D4    0.56
D5    0.76
D6   -0.04
D7   -0.04
D8   -0.06
D9    0.17
QU    1.00
```

tm パッケージ

- tm_0.2-3 by Ingo Feinerer
  - S4 クラスに基づく実装
  - 各種フォーマット・メタ情報 (XML, HTML, Gmane, RSS) への対応
  - 空白，stopwords の処理 (英独露など 13 言語に対応)
  - stemming の処理 (11 言語に対応)
  - 文章・ターム行列の作成
  - 各種重み付け (tf-idf など)

Feinerer: tmパッケージによる解析例

- テキストクラスタリング
  - Reuters-21578 データセットのサブテキスト (1720 文書)
  - bag of words：単語頻度情報
    - 古典的 k-mean 法 (`kmeans()`)

A.Karatzoglou & I. Feinerer: Text clustring with string kernels in R: Advances in Data Analysis, 2006.

H.Lodhi et al.: Text Classification using String Kernels: Machine Learning Reseach 2, 2002

Feinerer: tmパッケージによる解析例

- テキストクラスタリング
  - Reuters-21578 データセットのサブテキスト (1720 文書)
  - bag of words：単語頻度情報
    - 古典的 k-mean 法 (`kmeans()`)
  - String Kernels：文字の位置情報 (`stringdot()`)
    - **kernlab** パッケージによる kernel ベースの技法
    - Kernel k-means (`kkmeans()`)
    - Spectral Clustering (`specc()`)

A.Karatzoglou & I. Feinerer: Text clustring with string kernels in R: Advances in Data Analysis, 2006.

H.Lodhi et al.: Text Classification using String Kernels: Machine Learning Reseach 2, 2002

日本語テキスト

- 日本語テキスト解析
  - 文字解析
    - **grubsub** パッケージを利用

日本語テキスト

- 日本語テキスト解析
  - 文字解析
    - **grubsub** パッケージを利用
  - 形態素解析
    - 「すももももももものうち」

日本語テキスト

- 日本語テキスト解析
  - 文字解析
    - **grubsub** パッケージを利用
  - 形態素解析
    - 「すももももももものうち」
    - 茶筌や和布蕪などの形態素解析器との連携
    - すもも　も　もも　も　もも　の　うち
    - 名詞，助詞，...

和布蕪とのCインターフェイスの例

```
#include <Rdefines.h>
#include <Rinternals.h>
#include <mecab.h>
#include <stdio.h>
SEXP mecab(SEXP str){
    SEXP parsed;
    const char input = CHAR(STRING_ELT(str,0));
    mecab_t mecab;      mecab_node_t node;      const char result;
    mecab = mecab_new2 (input);
    result = mecab_sparse_tostr(mecab, input);
    PROTECT(parsed = mkString(result));      UNPROTECT(1);
    mecab_destroy(mecab);
    return(parsed);      }
```

和布蕪とのCインターフェイスの例

```
#include <Rdefines.h>
#include <Rinternals.h>
#include <mecab.h>
#include <stdio.h>
SEXP mecab(SEXP str){
    SEXP parsed;
    const char input = CHAR(STRING_ELT(str,0));
    mecab_t mecab;      mecab_node_t node;      const char result;
    mecab = mecab_new2 (input);
    result = mecab_sparse_tostr(mecab, input);
    PROTECT(parsed = mkString(result));      UNPROTECT(1);
    mecab_destroy(mecab);
    return(parsed);      }
```

R CMD SHLIB mecab.c でコンパイル後

```
    > dyn.load("mecab.so")
    > .Call("mecab", "すももももももものうち")
"すもも\t 名詞, 一般,*,*,*,*, すもも, スモモ, スモモ"
"も\t 助詞, 係助詞,*,*,*,*, も, モ, モ"
```

Rmecabの実装？

- R 上から日本語テキスト処理を一括して行う
  - n-gram などの文字カウント機能 (R の正規表現)
  - mecab へ形態素解析を委託
  - stopword の設定 (名詞のみ etc)
  - ターム文書行列の作成 (**lsa** の関数を利用)

Rmecabの実装？

- R 上から日本語テキスト処理を一括して行う
  - n-gram などの文字カウント機能 (R の正規表現)
  - mecab へ形態素解析を委託
  - stopword の設定 (名詞のみ etc)
  - ターム文書行列の作成 (**lsa** の関数を利用)
- yahoo のトピックス記事の分類
  - 防衛省人事問題，内閣改造，中華航空機事故
  - 検索語「防衛」との類似度測定