実践機械学習システム
3. クラスタリング：関連のある文書を見つける

河野 和平

Q&Aサイトの運営

- ユーザが探している情報について検索すると、サーチエンジンがそれに相応する「質問」と「回答」を提示する。

- 現在見ているページ内容と関連する情報（質問と回答）を提示したい。

単純なアプローチ

- 現在見ているページとそれ以外のページの類似度を算出
  - 類似度上位N個の文書のリンクを表示

- 計算量が膨大

レーベンシュタイン距離(1)

- 2つの単語があった場合に一方の単語をもう一方の単語に編集する最小回数

  「mchiene」→「machine」

  - レーベンシュタイン距離 2

    ①mの後にaを挿入：mchiene→machiene

    ②最初のeを削除：machiene→machine

レーベンシュタイン距離(2)

- 単語のテーベンシュタイン距離

  「How to format my hard disk」

  →「Hard disk format problems」

  - レーベンシュタイン距離 6

    ①How,to,format,myを削除

    ②format,problemsを文末に挿入

Bag-of-words

- 単語の出現回数を特徴量とする。

  ①How to format my hard disk

  ②Hard disk format problems

  ①{disk,format,how,hard,my,problems,to}

  ={1,1,1,1,1,0,1}

  ②{disk,format,how,hard,my,problems,to}

  ={1,1,0,1,0,1,0}

例題

Samples : 5

1	This is a toy post about machine learning. Actually, it contains not much interesting stuff.
2	Imaging databeses provide storage capabilities.
3	Most imaging databeses safe images permanently.
4	Imaging databases store data.
5	Imaging databases store data. Imaging databases store data. Imaging databases store data.

Features : 25

```
[u'about', u'actually', u'capabilities', u'contains', u'data', u'databases',
u'images', u'imaging', u'interesting', u'is', u'it', u'learning', u'machine',
u'most', u'much', u'not', u'permanently', u'post', u'provide', u'safe', u'storage',
u'store', u'stuff', u'this', u'toy']
```

{0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}

ベクトル化

- 新しい文書:「imaging databases」

  {0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}

  - 要素のほとんどが0であるベクトルになる(疎なベクトル)

- メモリを効率的に使用

  (0,7) 1

  (0,5) 1

ユークリッド距離(1)

- 新しい文書「Imaging databases」

  {0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
- 文書1

  {1,1,0,1,0,0,0,0,1,1,1,1,1,0,1,1,0,1,0,0,0,0,1,1,1}

  Dist = 4.0
- 文書2

  {0,0,1,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0}

  Dist = 1.73

ユークリッド距離(2)

- 新しい文書「Imaging databases」
  {0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
- 文書3
  {0,0,0,0,0,1,1,1,0,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0}
  Dist = 2.0
- 文書4
  {0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0}
  Dist = 1.41
- 文書5
  {0,0,0,0,3,3,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0}
  Dist = 5.10

正規化

- 文書4

  {0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0}

  Dist = 1.41
- 文書5

  {0,0,0,0,3,3,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0}

  Dist = 5.10

文書5は文書4を3回繰り返しただけの文書
にもかかわらず、ユークリッド距離が異なる。
→正規化によって統一する。

重要度の低い単語を取り除く

- 文書3:

  Most imaging databeses safe images permanently.

  - mostは分野に関係なく様々な文書で出現する(Stop word)
  - Imagesのような特定の分野で出現しやすい単語を重視すべき

文書3:{0,0,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,0}

新しい文書:{0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0}

Dist = 1.41

ステミング

- 意味的に同じ単語が語形変化によって異なる単語としてカウントされている
  - Imaging , images など
  - 同じ単語としてカウントする

TF-IDF

- これまでの方法は文書に特定の単語が何回出現するかを特徴量とした。
  - 文書中に特定の単語が多く存在すればするほど重要度が高くなる。

- どの文書にも出現する単語は重要度を低くすべき

$$TF-IDF = \frac{\text{対象の文書における単語}x\text{の出現回数}}{\text{単語}x\text{の出現する文書の数}}$$

クラスタリング

- フラットクラスタリング
  - クラスタ間の関係性は考慮しない
  - すべてのデータがどこか1つのクラスタに属する
  - 前もってクラスタ数を指定する必要がある

- 階層的クラスタリング
  - 類似性の高いデータをクラスタとしてグループ化
  - あるクラスタと類似するクラスタをまとめて親クラスタに分類
  - クラスタ数を指定する必要はない

KMeans

①クラスタの数を示すnum_clustersを指定

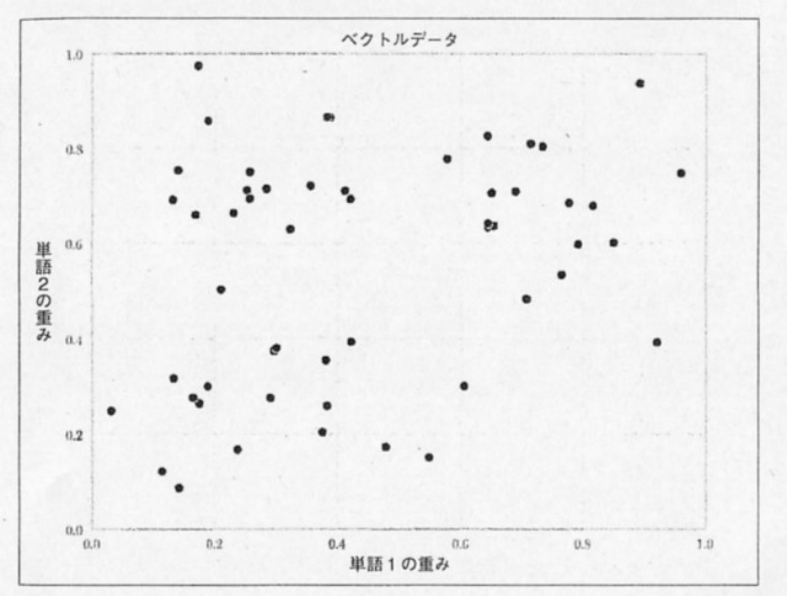
②num_clustersの数だけデータを任意に選ぶ

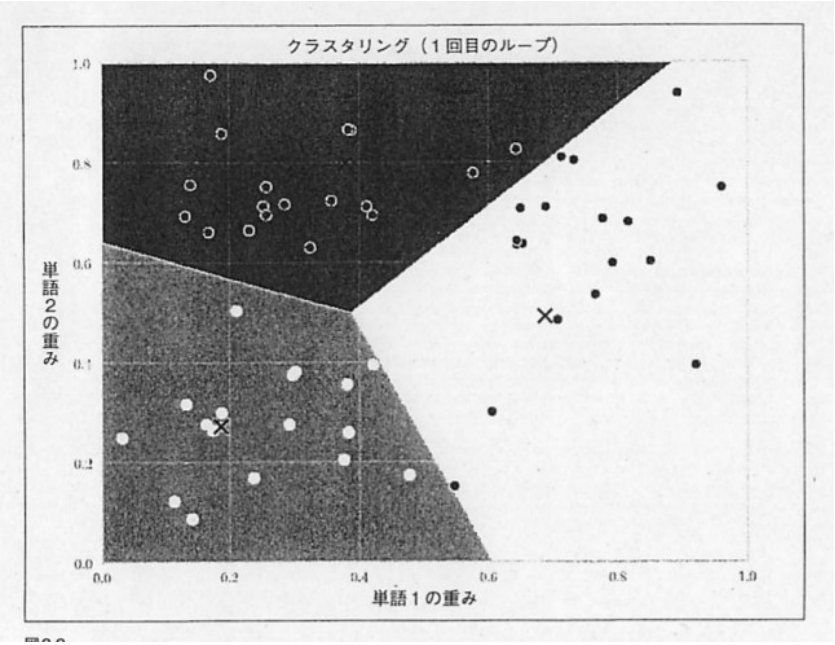
- その特徴ベクトルを中心点とする

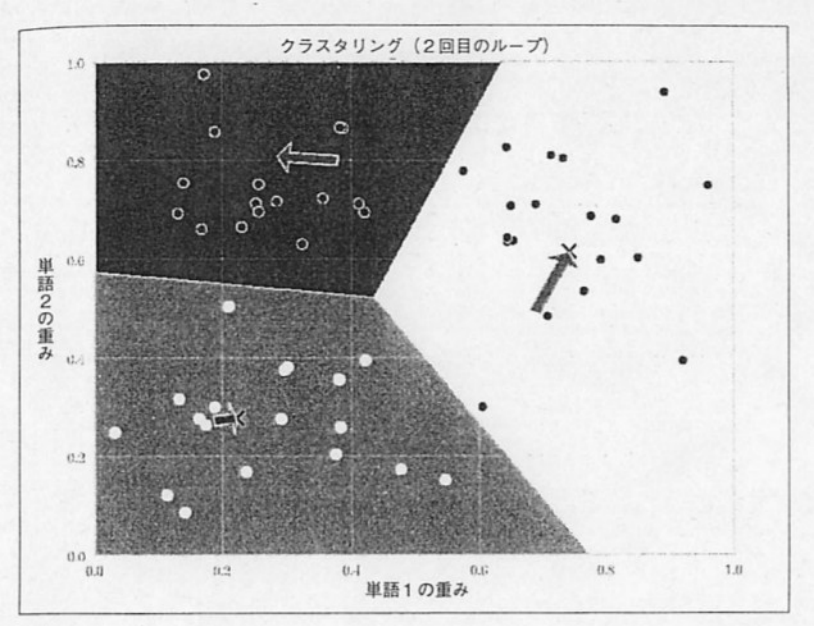
③残りのデータについて最も近い中心点を持つクラスタをそのデータのクラスタとする

④各クラスタの中心点を更新

（すべてのデータの平均ベクトルを中心点とする）


ベクトルデータ
単語2の重み
単語1の重み
0.0
0.2
0.4
0.6
0.8
1.0


クラスタリング（1回目のループ）
単語2の重み
単語1の重み
0.0
0.2
0.4
0.6
0.8
1.0


クラスタリング（2回目のループ）
単語2の重み
単語1の重み
0.0
0.2
0.4
0.6
0.8
1.0