集合知プログラミング
第6章　ドキュメントフィルタリング（1）
プログラム実行報告

05T4007T　江口晃

ドキュメントと単語～分類器

- 分類器に使う関数
  -テキストから単語を抽出する関数
  -単語、カテゴリ、ドキュメントをカウントする関数

- 分類器内のクラスの動作例

  入力データ

  ドキュメント ： カテゴリ
  'the quick brown fox jumps over the lazy dog' ： 'good'
  'make quick money in the online casino' ： 'bad'

  実行データ

  cl.fcount('quick' , 'good') ⇒ 1.0
  cl.fcount('quick' , 'bad') ⇒ 1.0

確率を計算する

- トレーニング（サンプル）データを作る

  ドキュメント　：　カテゴリ
  'Nobody owns the water'.　：　'good'
  'the quick rabbit jumps fences'　：　'good'
  'buy pharmaceuticals now'　：　'bad'
  'make quick money at the online casino'　：　'bad'
  'the quick brown fox jumps　：　'good'

- 追加する関数

  -単語が特定のカテゴリに存在する確率を求める関数

- 実行データ

  cl.fprob('quick' , 'good')　⇒　0.6666666666666663

推測を始める

- 追加する関数

  -仮の確率を使った確率を計算する関数（仮の確率=0.5、仮の確率の重み=1.0）

- 実行データ

```
>>>docclass.sampletrain(cl)
>>>cl.weightedprob('money' , 'good' , cl.fprob)
0.25
>>>docclass.sampletrain(cl)
>>>cl.weightedprob('money' , 'good' , cl.fprob)
0.16
```

単純ベイズ分類器

- 追加する関数

  -ドキュメント全体の確率（ドキュメント中の独立した単語の確率の積）を求める関数

  -ベイズの定理を使った確率（ドキュメント全体の確率×カテゴリの確率）を求める関数

- 実行データ

  cl.prob('quick rabbit' , 'good') ⇒ 0.15624999999999997

  cl.prob('quick rabbit' , 'bad') ⇒ 0.05000000000000003

カテゴリの選択

- 追加する関数
  - -しきい値をセット・取得する関数
  - -カテゴリを決定する（1番高いカテゴリの確率を探し、2番目に高い確率のしきい値以上か確認する）関数
- 実行データ

```
>>>docclass.sampletrain(cl)
>>>cl.classify('quick rabbit' , default='unknown')  ⇒  'good'
>>>cl.classify('quick money' , default='unknown')  ⇒  'bad'
>>>cl.setthresehold('bad' ,3.0)
>>>cl.classify('quick money' , default='unknown')  ⇒  'unknown'
>>>for i in range(10) : docclass.sampletrain(cl)
>>>cl.classify('quick money' , default='unknown')  ⇒  'bad'
```