実践　機械学習システム
第6章クラス分類II：感情分析

冨岡雄磨

感情分析とは

　書き手が取り扱っているテーマについての感情を調べることであり、**意見マイニング**とも呼ばれる

　簡易的に感情分析を行うとすると、

１．各単語にポジティブやネガティブなどのラベル付けを行った辞書を作成する

２．文章に出現した単語をラベルから値をつける

３．つけた値の平均値をとる

　という手順で算出できる

（文章を感情値化する処理を行う必要がある）

本章の概要

目的

Twitterに書かれた文章から感情分析を行いたい

分類を行うための分類機を作成する

目的に伴う問題点

・文字数の制限から、さまざまな略語が用いられる

・辞書に存在しない単語が用いられる

・感情が含まれないツイートが存在する

ツイートデータの取得

・Niek Sanders氏により作成された辞書を利用する（ツイートに対し手作業でラベル付けを行ったものである）

・ラベルは以下の4種類に分けられる

-意見なし

-ポジティブ

-感情なし

-ネガティブ

ただし意見なしと感情なしは同一のものとして扱う

ナイーブベイス分類器

　ベイス定理を用いた分類機であり、最も洗練された機械学習アルゴリズムのひとつ

　ベイス定理を用いるために、

「すべての特徴量は独立である」

という仮定が必要になる

　しかし、仮定が成立しない場合でも優れた性能を示すことが多い

ベイス定理

ベイス定理は次の式で表せます

$$P(A) \cdot P(B|A) = P(B) \cdot P(A|B)$$

P(A)　:単純にAが起こる確率
P(B|A):ある事象Aが起こる条件下で、
別の事象Bが起こる確率のこと。

変数	取りうる値	意味
C	“pos””neg”	ツイートが属するクラス（ポジティブかネガティブ)
F_1	非負整数	ツイートで「awesome」という単語が用いられた回数
F_2	非負整数	ツイートで「crazy」という単語が用いられた回数

もし、あるツイートについて、上の表のF_1とF_2の特徴量がわかっているとき、そのツイートがクラスCに属する確率を$P(C|F_1F_2)$と書くことができ、ベイスの定理を用いて変形すると、

$$P(C|F_1F_2) = \frac{P(C) \cdot P(F_1F_2|C)}{P(F_1F_2)}$$

と表すことができます

ベイス定理をナイーブに考える

まず、確率理論から次の式が成り立つ

$$\mathrm{P}(F_1F_2|\mathrm{C}) = \mathrm{P}(F_1|\mathrm{C}) \cdot \mathrm{P}(F_2|\mathrm{C}\,F_1)$$

ここで、「すべての特徴量は独立である」という仮定を行うと、

$$\mathrm{P}(F_1F_2|\mathrm{C}) = \mathrm{P}(F_1|\mathrm{C}) \cdot \mathrm{P}(F_2|\mathrm{C})$$

という簡単に式へ書きなおせる

これを利用すると、

$$\mathrm{P}(\mathrm{C}|F_1F_2) = \frac{\mathrm{P}(\mathrm{C}) \cdot \mathrm{P}(F_1|\mathrm{C}) \cdot \mathrm{P}(F_2|\mathrm{C})}{\mathrm{P}(F_1F_2)}$$

ナイーブベイスの計算

下の2つの式を計算し、より確率の高いクラスを選ぶ

$$\mathrm{P}(\mathrm{C}=\text{"pos"}\,|\,F_1F_2) = \frac{\mathrm{P}(\mathrm{C}=\text{"pos"}) \cdot \mathrm{P}(F_1|\mathrm{C}=\text{"pos"}) \cdot \mathrm{P}(F_2|\mathrm{C}=\text{"pos"})}{\mathrm{P}(F_1F_2)}$$

$$\mathrm{P}(\mathrm{C}=\text{"neg"}\,|\,F_1F_2) = \frac{\mathrm{P}(\mathrm{C}=\text{"neg"}) \cdot \mathrm{P}(F_1|\mathrm{C}=\text{"neg"}) \cdot \mathrm{P}(F_2|\mathrm{C}=\text{"neg"})}{\mathrm{P}(F_1F_2)}$$

右のような辞書があると仮定すると、
それぞれの値は以下のようになる

$\mathrm{P}(\mathrm{C} = \text{“pos”}) = 4/6 = 2/3$

$\mathrm{P}(\mathrm{C} = \text{“neg”}) = 2/6 = 1/3$

$\mathrm{P}(F_1 = 1|\mathrm{C} = \text{"pos"}) = 3/4$

$\mathrm{P}(F_1 = 0|\mathrm{C} = \text{"pos"}) = 1/4$

$\mathrm{P}(F_2 = 1|\mathrm{C} = \text{"pos"}) = 2/4 = 1/2$

$\mathrm{P}(F_2 = 1|\mathrm{C} = \text{"neg"}) = 2/2 = 1/1$

$\mathrm{P}(F_1 = 1, F_2 = 0) = 2/6 = 1/3$

ツイート	クラス
awesome	ポジティブ
awesome	ポジティブ
awesome crazy	ポジティブ
crazy	ポジティブ
crazy	ネガティブ
crazy	ネガティブ

F_1=awesome
F_2=crazy

問題点の解決(1)：新出単語への対応

　もし、「text」というツイートが与えられたとき、先ほどの辞書では対応できない。このような問題に対して行われる対応が
「１を足すスムージング」である（加算スムージングとも呼ばれる）

　すべての頻度に1を足すだけであり、「その単語を辞書がたまたま含んでいなかっただけである」というチャンスを与えていることを仮定している

この際の計算は、

$P(F_1 = 1 | C = \text{“pos”}) = 3/4$　　ではなく、

$P(F_1 = 1 | C = \text{“pos”}) = 3+1/4+2 = 4/6$

ここで分母に2を足しているが、これは確率の体裁を維持するためであり、

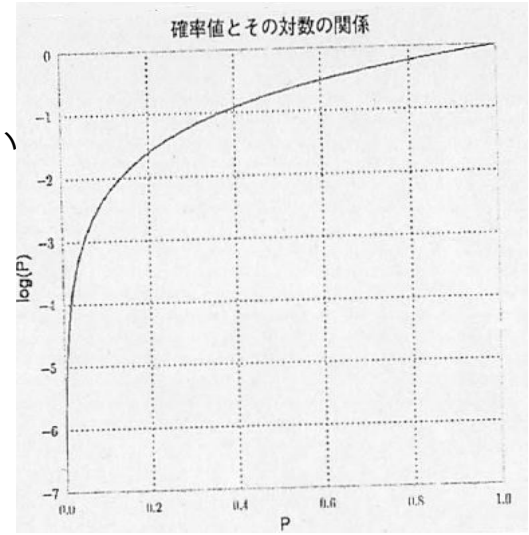
$P(F_1 | C = \text{“pos”}) = P(F_1 = 0 | C = \text{“pos”}) + P(F_1 = 1 | C = \text{“pos”}) = 2/6+4/6 = 1$

となる

問題点の解決(2)：アンダーフローへの対応

　アンダーフローとは、計算結果の値が一定以上小さくなると0として処理される現象であり、数千や数万単語を扱うと予想されるこのシステムでは十分に起こりえる

　最も簡単な対処法としては、「対数を使う」というものがあげられる
　ｘ・ｙ＝log(x)+log(y)ですので、指数関数的に
値が小さくなるということが無くなるだけでなく、
0～1をとっていたものが-∞～0をとるようになるため、
値が大きいほど可能性が高いという評価式も変わらない



　このときの計算式は

$$\log P(C) + \log P(F_1|C) + \log P(F_2|C)$$

であらわせる

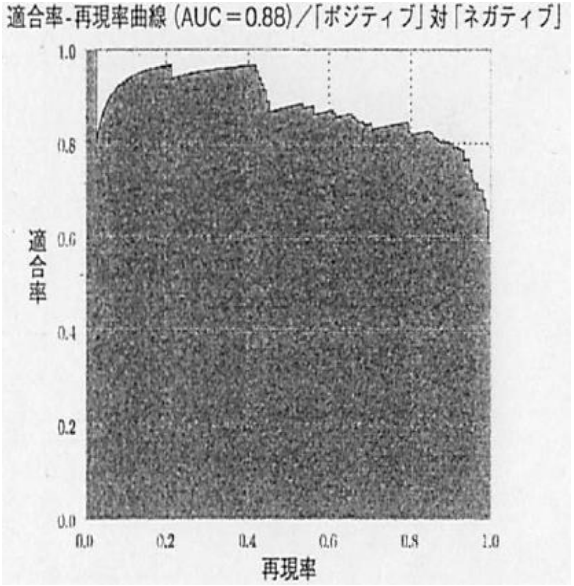
機能の調整：すべてのクラスを対象とする

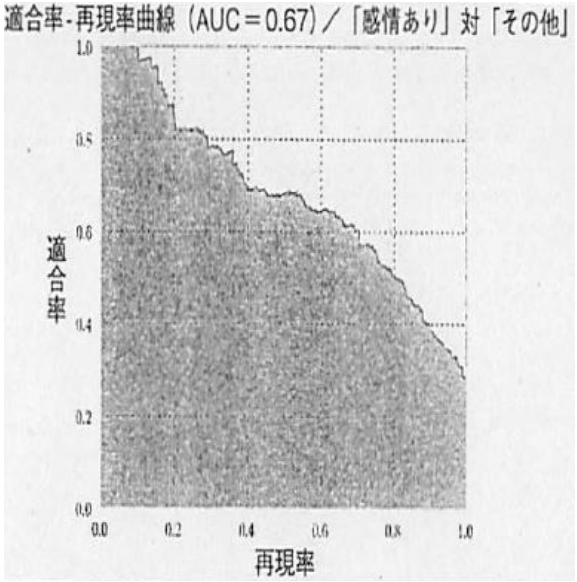
　これまで作成してきたものはポジティブとネガティブの分類器である

　しかし、最初に予定していた分類器は4つのクラスを分類するものであった

　これを達成するために、感情あり（ポジティブとネガティブ）と感情なし（意見なしと感情なし）の分類器を作成し、ポジティブとネガティブの分析器と連動させる

　1と-1という反対のものから、1と0や-1と0というより近い感情の判別のためか、ポジティブとネガティブの判定と比べ、ＡＵＣが大きく下がっている





機能の調整：パラメータの調整

　分類器のパラメータを調整することでより正確に分類できるようになることがあります。しかし、すべてを検証するには大きな手間がかかります。

　そのため、GridSearchCVと呼ばれる専用のクラスで総当りでパラメータを探します。

操作できるパラメータ一覧

- **TfidfVectorizer**
  - NGramsについて次の場合を試します：ユニグラム (1,1)、バイグラム (1,2)、トライグラム (1,3)
  - min_dfについては、1または2の場合を試します。
  - TF-IDFにおけるIDFの影響を検証するため、use_idfとsmooth_idfについてFalseとTrueの場合を試します。
  - stop_wordsにEnglishまたはNoneを設定し、ストップワードを用いる場合と用いない場合を検証します。
  - 単語の頻度（sublinear_tf）について対数を用いるかどうかを検証します。
  - 記録する対象を単語の出現回数（頻度）か単語の出現の有無にするかを、binaryをTrueまたはFalseにして試します。
- **MultinomialNB**
  - スムージングについてを検証するため、alphaに次の値を設定します。
  - ラプラス・スムージング（1を足すスムージング）：1
  - Lidstoneスムージング：0.01、0.05、0.1、0.5
  - スムージングなし：0

機能の調整：言葉の意味の追加

以下のようなことを行うことで、より性能を上げることができる

・略語や顔文字などを辞書に登録する

・単語の種類などの言語学を用いた情報の追加

例：感情が含まれないツイートには名詞が多く、形容詞や動詞が多いほど感情が含まれる

・SentiWordNetを活用する

ほとんどの英単語についてポジティブやネガティブのパラメータが設定されているだけでなく、同義語ごとに異なる値が設定されている

一言で言えば、情報が増えれば精度は上がる

まとめ

・ナイーブベイスは、ベイス定理において「すべての特徴量は独立である」と仮定したものであるが、そうでないものに対しても力を発揮する

・パラメーターの調整や情報の追加といったものだけでも精度を大きく上げることができる

```
== Pos vs. rest ==
0.866    0.010    0.327    0.017
== Neg vs. rest ==
0.861    0.010    0.560    0.020
```

正答率 +2%
AUC +20.6%

正答率 +2%
AUC +8.9%

```
== Pos vs. rest ==
0.886    0.006    0.533    0.026
== Neg vs. rest ==
0.881    0.012    0.629    0.037
```