言語処理のための機械学習入門

2.4～2.6

茨城大学工学部情報工学科
08T4018Y　小幡智裕

2.4 文書に対する前処理と データスパースネス問題

2.4.1 文書に対する前処理

- ストップワード(stopword)
  文書中の ”the”, ”is”, “have”, “take”など、
  話題の種類と関連を持たないと考えられる語。
  ストップワードをあらかじめ削除してからベクトル化を行うことが多い。

- ステミング(stemming)
  同じような話題を指している語を一つの素性としてみなしたいとき、
  派生語なども含めて同一の素性とみなす作業のこと。

  例. ”run”, “runs”, “ran” さらに”runner”など。

2.4.1 文書に対する前処理（続き）

・ポーターのステマー(Porter’s stemmer)

英語のステミングの手法。多くの規則がある。

規則の例

- 語尾のedを除去する(walked→walk)
- 語尾のateを除去する(passionate→passion)
- 語尾のationalを除去する(operational→oper)

“operational”と ”operate”は同じ ”oper” として扱えるが、
(hundred→hundr)となってしまったり、
(international→intern)と、意味が変わってしまうこともある。

2.4.1 文書に対する前処理（続き）

- 見出し語化(lemmatization)

  “run”, “runs”, “ran” などの同じ単語が変化しただけのものを同一の素性とみなしたいとき、すべての単語を基本形に戻す作業のこと。
  見出し語化は、文脈（周囲に出現する単語等）を考慮して行われる。

- 語義の曖昧性解消(word sense disambiguation)
  同音異義語を区別すること。

  例. fly[動詞]→飛ぶ、 fly[名詞]→ハエ
  品詞によって区別し、異なる素性として扱う。
  品詞まで判定することで見出し語化も可能となる。

  bank[名詞]→銀行、 bank[名詞]→土手
  品詞による単語の区別を行っても単語の意味の区別ができない。

2.4.2 日本語の前処理

- 形態素解析(morphological analysis)

  単語の区切りを判定することを**単語分割**という。

  同時に**品詞タグ付け**も行われることが多い。

  この二つの作業のことを**形態素解析**という。

  日本語だけでなく、中国語、タイ語などでも行う必要がある。

2.4.3 データスパースネス問題

一般に、0でない値をとる要素数が小さい傾向があるとき、このデータは**疎(sparse)**であるという。

データが疎である場合、そのデータを処理するために必要な統計値が充分に獲得できないことがあり、この問題を一般的に

データスパースネス問題(data sparseness problem)という。

2.5単語のベクトル表現

・単語が含む文字を用いてベクトルを作る

例.quicklyを文字バイグラムで表現する

→qu, ui, ic, ck, kl, lyに対応する要素だけ

1となり、それ以外が0となるベクトルができる

・単語 をコーパスで探す。

w の直前や直後に出現する単語群は w を特徴づけている。この単語群が w に対応する文書であるかのようにして w をベクトルを表わすことができる。

→本節で紹介

2.5.1 単語トークンの文脈ベクトル表現

例文. 高く **跳ぶ** には まず 屈め .

$$x_{\text{跳ぶ}} = (n(\text{"高く"}), n(\text{"に"}), n(\text{"は"}), n(\text{"まず"}), n(\text{"屈め"}))$$

$$= (1,1,0,0,0)$$

文脈から作られている。

→**文脈ベクトル(context vector)**という。

対象単語の前後の数トークンを考慮しているとき、文書中の考慮している箇所を **文脈窓(context window)** といい、またその大きさを **文脈窓幅(context window size)** という。

2.5.1 単語トークンの文脈ベクトル表現（続き）

文脈窓内のトークンを位置によって区別する。

例文. 危険 を **恐れ** ず 攻め よ

2トークン前における”危険”を”危険”$_{-2}$とする

$$x_{\text{恐れ}} = (n(\text{"危険"}_{-2}), n(\text{"危険"}_{-1}), n(\text{"危険"}_{+1}), n(\text{"危険"}_{+2}), \ldots)$$
$$= (1,0,0,0,\ldots).$$

構文的な情報を用いてベクトルを作ることもできる。

→文脈から作るより詳細に構文的な振舞いを表現できる。

しかし、時間がかかる。

2.5.2 単語タイプの文脈ベクトル表現

複数の文脈窓内で、どんな単語が何回出現したかを素性とする。

例文. “Nothing ventured, nothing gained.”

$$x_{nothing} = (n("\text{ventured}"_{+1}), n("\text{ventured}"_{-1}), n("\text{gained}"_{+1}),$$
$$n("\text{gained}"_{-1}), n(",")_{+1}), n(",")_{-1}))$$
$$= (1,0,1,0,0,1).$$

ここでは、”,”も1単語として考えた。

実際の語彙はもっとずっと大きいため、0がたくさん並んだベクトルとなる。

2.6 文書や単語の確率分布による表現

- 文書

確率分布 $P(W \mid d)$ を考え、これが文書を表わしているとみなす。
w は各単語を値とする確率変数で、d は与えられた文書である。

確率分布の決め方は、最尤推定で $P(W \mid d) = n_w / \sum n_w$
とするのがもっとも単純である。

2.6 文書や単語の確率分布による表現（続き）

- 単語タイプ

単語タイプ w が出現した条件のもとで周囲に単語タイプ V が出現する、条件付き確率 $P(V \mid w)$ が単語タイプ w を表現しているとみなす。

V は w の周囲に出現する単語タイプに対応する確率変数を表わしている。

例. “Nothing ventured, nothing gained.” を用いて ”nothing”を確率の言葉で表現する。パラメータは最尤推定で求める。

$$P(V = \text{"ventured"}_{+1} \mid w) = 0.33, P(V = \text{","}_{-1} \mid w) = 0.33,$$
$$P(V = \text{"gained "}_{+1} \mid w) = 0.33$$

となる。他の V については、$P(V = v \mid w)$ は0である。