# 確率的言語モデル

# 演習３
# 最大エントロピーモデル(対数線形モデル)による単語分割

吉野 幸一郎

2011/7/28

# 目次

- 単語分割とは
- SVMを利用した単語分割
- 確率的単語分割
- MEモデル(＝対数線形モデル)
- 対数線形モデルによる単語分割

# 日本語単語分割

- 日本語や中国語には単語境界がない
- 単語単位でコーパスを扱うための必須のタスク

**日 本 語 や 中 国 語 の よ う に**

**日-本|語|や|中-国|語|の|よ-う|に**

- 各文字間に -(境界なし) or |(境界あり) のどちらが入るかを決定する問題

- 詳しくは
- http://www.phontron.com/kytea/method-ja.html
- http://www.ar.media.kyoto-u.ac.jp/members/yoshino/tutorial/word-seg.html

# 単語分割に利用できそうな素性

- 自身の文字 & 前後の文字n-gram
  - 前後にどのような文字の分布があるか

**日-本|語|や|中-国|語|の|よ-う|に**

- 自身の文字種 & 前後の文字種n-gram
  - 前後にどのような文字種の分布があるか

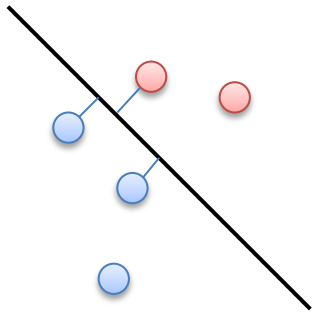
**最-初|の|タ-ス-ク|で|あ|る|。**

- 前後の単語n-gram
  - 分割時にわかるのは「前の単語の推定値」のみ

- CRF等で「最適な単語境界ラベル列を解く問題」として解くこともできます

# SVMを利用した単語分割

- SVMとは
  - N次元空間においてマージン最大化という考え方に基づいて分離超平面を引くもの
- SVMは2クラス分類器
  - 単語分割は2値分類問題＋スパースな素性(高次元のベクトル)を使うのでSVMが適している



- Kyteaでは推定値の情報を利用せず実データをpointwiseに利用するため、容易な拡張が可能
  - 利用する素性は「文字n-gram」「文字種n-gram」のみ

- しつこいようですが詳しく知りたい場合は先述のURLを参照するとよいです

# 確率的単語分割

- 各文字間に文境界がある確率(|が入る確率)を計算する
  →各文字が「単語末尾である確率$P_e(c)$」を計算する

$$s = (w_1, w_2, \dots, w_{n-1}, w_n)$$

$$\begin{aligned} P(s) &= P(w_1, w_2, \dots, w_{n-1}, w_n) \\ &= P(\underbrace{c_{b_1}, c_{b_2}, c_{e_1}, \dots}_{w_1}, \underbrace{c_{b_k}, c_{e_n}}_{w_n}) \\ &= \left(1 - P_e(c_{b_1})\right)\left(1 - P_e(c_{b_2})\right) P_e(c_{e_1}) \times \cdots \times \left(1 - P_e(c_{b_k})\right) P_e(c_{e_n}) \end{aligned}$$

# 確率的言語モデルに基づいた単語分割

- log(P(農産物価格安定法)) = -23.8705
- log(P(農 産物価格安定法)) = -27.9562
- log(P(農産 物価格安定法)) = -25.3834
- log(P( 農 産 物価格安定法)) = -31.3807
- log(P(農産物 価格安定法)) = -26.684399
- …
- log(P( 農 産物 価格 安定 法)) = -28.530602
- log(P(農産 物 価格 安定 法)) = -23.044102
- log(P( 農 産 物 価格 安定 法)) = -32.564503

- 分割候補を列挙し全候補に対し言語モデルから確率を与える
- 最も確率が高くなる候補を選ぶ
- 計算のオーダーが$O(2^{n-1})$になる(nは文字数)
- DPを使うと効率的に計算できる

# 動的計画法による1-gram単語分割

- 例文:農産物価格安定法

1. $P$(農)を計算する
2. $P$(農)$P$(産)と$P$(農産)のmaxを$P$(農産)とする

…

- 各文字でmaxになったものだけを保持して分割をする

# Maximum Entropy Modelとは

- 今回の問題
  - 素性ベクトル $X$ と射影されるクラス $y$ があったとき
  - $p(X, y)$ が問題 → $X$ がgivenで $y$ を考える
    → 条件付確率 $P(y|X)$ を考える

- Maximum Entropy Modelは、「素性関数以外の制約は与えられていない（不確か）なのでそれ以外に関してはエントロピーを最大とするような一様分布に従うべき」という考え方に基づく

- エントロピーを最大化＝一様分布との距離を最小化

# エントロピー

- エントロピーの定義

$$H(p) = \frac{\sum_s -\log_2 P(s)}{\text{コーパスの文字数}}$$

言語モデルでの定義

$$H(p) = -\sum_{x \epsilon X} p(x) \log p(x)$$

- 条件付きエントロピー

$$H(p) = -\sum_{x \epsilon X} p(x) \sum_{y \epsilon f(x)} p(y|x) \log p(y|x)$$

- これを
  - 条件付き確率の和が1
  - 期待値が経験分布と真の分布で一致
- という制約下でラグランジュの未定乗数法で解く…と

# エントロピー

- ざっくり省略して
  - 気になる人は下のURL辿れば導出書いてあります

$$p(y|x) = \frac{1}{\sum_y \exp\left(\sum_j \lambda_j f_j(x,y)\right)} \exp\left(\sum_j \lambda_j f_j(x,\ y)\right)$$

とか出てくる。これは対数線形モデルの

$$p(y|x) = \frac{1}{\sum_y \exp(w\varphi(x,y))} \exp(w\varphi(x,y))$$

の式と形が同じ

- Maximum Entropy Model = log-linear Model (ロジスティック回帰も)
- http://www.r.dl.itc.u-tokyo.ac.jp/~ninomi/mistH21w/cl/mistH21w-ninomi-12.pdf

# 対数線形モデルによる確率的単語分割

- 素性列 $X$ が与えられたとき $y$ が単語末尾である確率$P(y|X)$をモデル化する
- 素性としては前後の文字2-gram・文字種2-gramを用いる
  - 今回用いる素性

```
2 Lc1=BT:1 Cc1=日:1 Rc1=本:1 Lt1=BT:1 Ct1=C:1 Rt1=C:1
2 Lc1=日:1 Cc1=本:1 Rc1=語:1 Lt1=C:1 Ct1=C:1 Rt1=C:1
1 Lc1=本:1 Cc1=語:1 Rc1=や:1 Lt1=C:1 Ct1=C:1 Rt1=H:1
1 Lc1=語:1 Cc1=や:1 Rc1=中:1 Lt1=C:1 Ct1=H:1 Rt1=C:1
2 Lc1=や:1 Cc1=中:1 Rc1=国:1 Lt1=H:1 Ct1=C:1 Rt1=C:1
2 Lc1=中:1 Cc1=国:1 Rc1=語:1 Lt1=C:1 Ct1=C:1 Rt1=C:1
1 Lc1=国:1 Cc1=語:1 Rc1=の:1 Lt1=C:1 Ct1=C:1 Rt1=H:1
```

# 素性関数

- 対数線形モデルにおいては素性関数を定義し、素性関数によってベクトルを構成する

- $\varphi_k(X, y) = \begin{cases} 1 \ \ if \ \text{ある素性が} X \text{で出現} + \text{そのラベルが} y = 1 \\ 0 \qquad\qquad\qquad else \end{cases}$

- ベクトルを作って正解ラベルを振るのではなく正解ラベルとベクトルを組み合わせたベクトルを作る(SVMとの違い)

  - $\varphi_1(X, y) = \delta(\text{"Lc1=BT"} \in X)\delta(y = 1)$
  - $\varphi_2(X, y) = \delta(\text{"Lc1=BT"} \in X)\delta(y = 2)$

    …

- というような関数を延々作る

# 素性関数の重み

- $w\varphi(X,y)$ とは…
  - $w\varphi(X,y) = (w_1\varphi_1(X,y), w_2\varphi_2(X,y), \ldots, w_n\varphi_n(X,y))$
  - 定義した素性関数分重み$w$を推定する必要がある
  - 推定のためには対数尤度を用いる
- $w$の意味
  - $w$は「その素性関数がどれだけ効くか」つまり、「その素性がそのクラス分類にどれだけ寄与するか」を示す
  - 分類の際未知の素性があった場合、$w = 0$となるのでその素性関数は無視される(分類に寄与しない)ことになる

# 対数線形モデルによる確率的単語分割

- 学習
  - 対数線形モデルは $P(y|X)$ をモデル化するのでこの対数尤度を考える
  - 先程の関数ごとの重みを推定するということ

$$\mathrm{L}(w) = \sum_{X^{(i)},y^{(i)}} \log P(y^{(i)}|X^{(i)}) - \frac{C}{2}|w|^2$$

$$= \sum_{X^{(i)},y^{(i)}} \log\exp\left(w\varphi\left(X^{(i)},y^{(i)}\right)\right) - \log\sum_{y^{(i)}} \exp(w\varphi(X^{(i)},y^{(i)})) - \frac{C}{2}|w|^2$$

$$= \sum_{X^{(i)},y^{(i)}} w\varphi\left(X^{(i)},y^{(i)}\right) - \log\sum_{y^{(i)}} \exp(w\varphi(X^{(i)},y^{(i)})) - \frac{C}{2}|w|^2$$

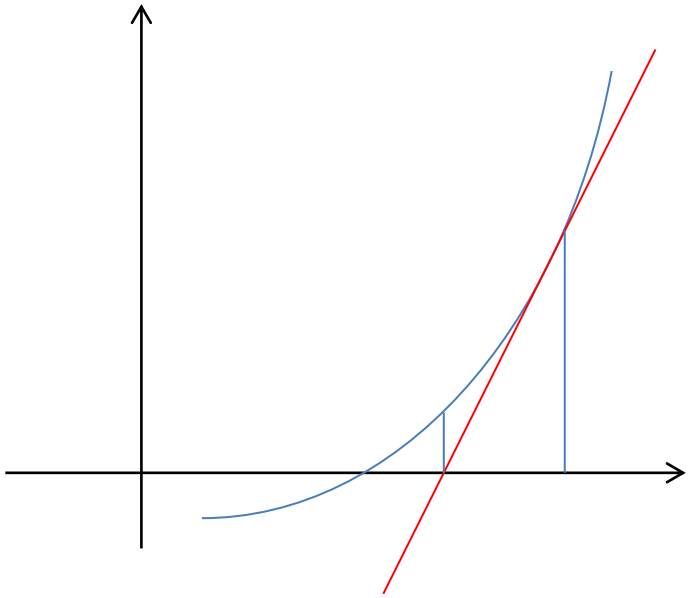
  - 最急勾配法かニュートン法を使う
  - 今回はニュートン法

# ニュートン法

- 関数を１回微分すると、微分後の関数とx軸との交点が元の関数の極値であることを利用する
- 関数を２回微分すると、変曲点を含まない限りにおいては２回微分の関数とx軸の交点は1回微分の交点に収束する

- これの性質を利用して

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$



- をひたすら繰り返す
- (ある程度収束したら
  適当なところで止める)

# 実装の手順

- SVMで定義したベクトルと正解ラベルを組み合わせて素性ベクトルを作るための素性関数を作る
- 各素性関数 $\varphi_k(X, y)$ ごとの重み $w$ を学習する
- 分類するときは分類事例の正解ラベルが1,2両方の場合の素性ベクトルを作り、重み $w$ の列との内積を取ることによって分類ができる（内積が高い方のクラス）
- $\max_y \exp(w\varphi(X, y))$
- から$y$のラベルを推定する

# 課題

選択式

1. 対数線形モデルを実装し、MPTコーパス200文を学習セットとして65文の分割精度を測定する

2. MPTコーパス265文から1-gram確率モデルを学習しDPを実装して単語分割をやってみる
   - 実行時間と例文・分割結果を書く

3. MPTコーパス265文から2-gram確率モデルを学習しDPを実装して単語分割をやってみる
   - 実行時間と例文・分割結果を書く

# 参考文献


- 言語処理のための機械学習入門　高村大也
- 数理言語情報論　第12回　二宮崇
  - http://www.r.dl.itc.u-tokyo.ac.jp/~ninomi/mistH21w/cl/mistH21w-ninomi-12.pdf
- LIBLINEARを用いた機械学習入門
  - http://www.ar.media.kyoto-u.ac.jp/members/yoshino/tutorial/word-seg.html
- Kytea
  - http://www.phontron.com/kytea/method-ja.html