言語処理における識別モデルの発展 – HMMからCRFまで

坪井 祐太　　鹿島 久嗣　　工藤 拓
日本IBM株式会社 東京基礎研究所　　グーグル株式会社

1　はじめに

自然言語処理の分野では, 配列や木, グラフなどの構造を持ったデータを扱うような問題が多く見られる. 特に, 構造データから構造データへのマッピングを行う問題として定式化できる課題が多い.

たとえば固有表現抽出を考えてみよう. 固有表現抽出は文書中に含まれている人名・地名・組織名等を特定する問題である. 固有表現抽出は以下の様に単語列に対して固有表現の始まり (B-XXX) と続く固有表現 (I-XXX) を示す目的ラベルを付与する問題として考えることができる[1]. 次の例では「日本」を地名 (LOC), 「小泉内閣」を組織名 (ORG) として抽出する様を表したものである.

単語列 $\boldsymbol{x}$	日本	で	小泉	内閣	が	発足し	た	.
ラベル列 $\boldsymbol{y}$	B-LOC	O	B-ORG	I-ORG	O	O	O	O

ただし, O は固有表現以外を意味する. 同様にして, 品詞付与もこのような問題として捉えることができる.

この問題をマッピングの問題として捉えると, 単語の列から, ラベルの列へのマッピングを行っていると考えることができる. 図 2(a) は単語列に対するラベル付与を図示したものである. また, 何も単語列に限らず, 単語間の係り受けを表現した木やグラフなどの構造をもった文に対しても同様に定義できる. 図 2(b) は文法的な単語間の係り受け関係を示した依存構造木に対するラベル付与を図示したものである.

また, もっと複雑な例として, 構文解析を考えてみよう. この場合, 入力は単語の列であるが, 出力が単語列の上に構成される構文解析木となる. 先ほどの問題と異なり, 単に配列や木, グラフ中のノードに対してラベルを振るだけでなく, 出力の構造自体を構成する必要があるため, より複雑な問題となる.

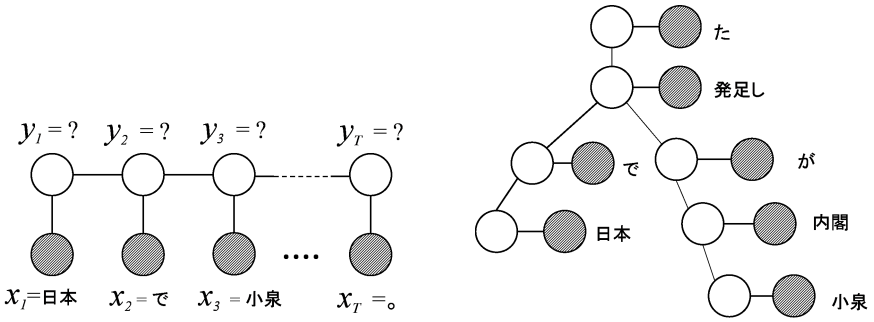
本稿で紹介するモデルは, これらの問題をまとめて扱うことができる汎用的なモデルであるが, 説明を簡単にするために, 本稿では, 前者のラベル付け問題, すなわち出力の構造はあらかじめ固定されており, ノードのラベル付けのみを行う問題のみを扱うことにする.

さて, この問題をもう少し形式的に定式化してみよう. 構造データの入力にあたる部分を $\boldsymbol{x}$(先の例における単語列), 出力にあたる部分を $\boldsymbol{y}$(先の例における固有表現ラベル) と書くことにする. ここで実現したいことは, $\boldsymbol{x}$ が与えられたときに, 対応する $\boldsymbol{y}$ を正しく出力することである. この対応関係をルールとして書き出すのもひとつの方法ではあるが, 近年では, 適切な $\boldsymbol{x}$ と $\boldsymbol{y}$ のペアを学習データ (過去の事例) として, ここから統計的な推論によって対応関係を導こうというアプローチが広く受け入れられている. 従って, 与えられた学習データから, 何がしかの関数 $X \rightarrow Y$ を学習するということを行うことになる. (ここで X はすべてのありうる入力データの集合, Y はすべてのありうる出力データの集合であるとする.) この関数が正しく求められれば, 任意の入力 $\boldsymbol{x}$ に対し, 適切な $\boldsymbol{y}$ を出力することができるはずである.

しかし当然のことながら, すべての正しい $\boldsymbol{x}$ と $\boldsymbol{y}$ が与えられるはずもなく, この関数は限られた学習データから求める必要がある. また, 未知データに対して 100%予測可能なモデルをつくることができるとは考えにくく, 必然的に予測は確率的なものとなる. では, どのように確率的であればよいだろうか? $\boldsymbol{x}$ から $\boldsymbol{y}$ を予測するための確率分布としては, 条件付確率分布 $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を考えればよいだろう. これがあれば, $\boldsymbol{x}$ が与え

[1] この形式は IOB2 と呼ばれる固有表現のラベル方法で, その他様々な記述方法が提案されている [3].

図 1: グラフ構造の例: 斜線のノードは入力変数 $\boldsymbol{x}$ を, 白抜きのノードが出力変数 $\boldsymbol{y}$ を示す.



(a) 単語列　　(b) 依存構造木

られたときに, $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を用いて, 予測 $\hat{\boldsymbol{y}}$ を

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in Y} \Pr(\boldsymbol{y}|\boldsymbol{x})$$

と決めればよいことになる. ここで, $\operatorname*{argmax}_{\boldsymbol{y} \in Y} \Pr(\boldsymbol{y}|\boldsymbol{x})$ は,「すべての出力候補 Y のなかでもっとも大きな $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を実現するような $\boldsymbol{y} \in Y$」を意味するとする.

さて, 枠組みは決まったが, より話を具体化していくうえで考えるべき点は, 主に次の 2 点である.

問題 1: $\Pr(\boldsymbol{y}|\boldsymbol{x})$ をどのような形にすればよいか?
簡単すぎず, 複雑すぎず, 問題の普遍的な特徴をうまくとらえた, 予測力の高いモデルを設計する必要がある. $\Pr(\boldsymbol{y}|\boldsymbol{x})$ が, パラメータ θ によって決まるとするとき, これを $\Pr(\boldsymbol{y}|\boldsymbol{x};\theta)$ と書くことにすると, パラメータ θ をどのように定義すればよいか, また $\Pr(\boldsymbol{y}|\boldsymbol{x};\theta)$ をどのような形にすればよいかを考えなければならない.

問題 2: 与えられたデータから, どのように $\Pr(\boldsymbol{y}|\boldsymbol{x};\theta)$ を求めればよいか?
データの情報を最大限に利用するためには, 何かの基準のもとにモデルパラメータを決定する必要がある. つまり, より良い予測力に結びつくような基準 (＝目的関数) を考え, これを学習データを用いて最適化, つまり最適なパラメータ θ を決定する必要がある. 当然ながら, このパラメータの決定も正しく, 効率的に行える必要がある.

本稿で紹介するモデルは, これらの問題に対するアプローチの違いから説明することができる.

以下の章では, まず, これらのタスクにおいて伝統的に用いられてきた隠れマルコフモデル (HMM) を紹介し, 次に, 近年, 重要視されつつある識別モデルに基づく確率モデルである, 条件付確率場 (CRF) を紹介し, 生成モデルと識別モデルの違いという立場からこれらの違いを考える. また, 日本語形態素解析タスクでの実験を通じてこれを検証してみることにする. 最後に, 公開されている CRF のツールをいくつか紹介する.

2　生成モデル：隠れマルコフモデル (HMM)

配列に対してのラベル付与問題では, 古くより隠れマルコフモデル (HMM)[2] が用いられ，さまざまなタスクにおいて成功を収めてきた.

HMM ではまず, 観測変数と目的変数の同時確率 $\Pr(\boldsymbol{x}, \boldsymbol{y})$ を考えるところからはじまる. これは, 文 $\boldsymbol{x}$ とラベル $\boldsymbol{y}$ を生成するような確率分布の存在を仮定することから, HMM は生成モデルといわれる. 図 2(a) のような配列のラベル付け問題について, 具体的に HMM のモデルを考えてみる.

入力変数の集合と, 出力変数の集合をそれぞれ左から $\boldsymbol{x} = (x_1, x_2, \ldots, x_T), x_t \in \Sigma_x$ と, $\boldsymbol{y} = (y_1, y_2, \ldots, y_T), y_t \in \Sigma_y$ とする. 形式的に, 左端と右端に特別なラベルをとる仮想的な変数 y_0 と y_{T+1} があるものとしよう.

これらの変数を用いて, $\Pr(\boldsymbol{x}, \boldsymbol{y})$ は以下のように定義される.

$$\begin{aligned} \Pr(\boldsymbol{x}, \boldsymbol{y}) &= \theta_{y_0,y_1}\theta_{y_1,x_1}\theta_{y_1,y_2}\theta_{y_2,x_2}\cdots\theta_{y_{T-1},y_T}\theta_{y_T,x_T}\theta_{y_T,y_{T+1}} \\ &= (\prod_{t=1}^{T}\theta_{y_{t-1},y_t}\theta_{y_t,x_t})\theta_{y_T,y_{T+1}} \end{aligned} \tag{1}$$

ここで, $\theta_{y_t,y_{t+1}}$ や θ_{y_t,x_t} をまとめてモデルのパラメータ θ であるとし,

$$\theta_{y_t,y_{t+1}} = \Pr(y_{t+1}|y_t) \text{ および } \theta_{y_t,x_t} = \Pr(x_t|y_t)$$

であるとする. つまり, HMM では y_t は直前の y_{t-1} に, x_t は y_t のみに依存して決まるとすることで, 配列の性質は, 連続する 2 つの出力変数の組 (y_{t-1} と y_t) と, 同じ位置での出力変数と入力変数の組 (y_t と x_t) で, 局所的な構造の積み重ねによって表現されるということを仮定しているのである. なお, HMM が確率モデルになっていることを保証するために, 任意の $y \in \Sigma_y$ に対して,

$$\sum_{y' \in \Sigma_y} \theta_{y,y'} = 1 \text{ および } \sum_{x \in \Sigma_x} \theta_{y,x} = 1 \tag{2}$$

であることに注意する.

ここで少し寄り道をして, HMM の確率モデル (1) を別の表現で書いてみることにする. ある $(\boldsymbol{x}, \boldsymbol{y})$ において, $y \in \Sigma_y$ のあとに $y' \in \Sigma_y$ が現れた回数を $\phi_{y,y'}$ とおく. 同様にして, $y \in \Sigma_y$ と同じ位置に $x \in \Sigma_x$ が現れた回数を $\phi_{y,x}$ とおく. これらの表記を用いると, (1) は次のように書き直すことができる.

$$\Pr(\boldsymbol{x}, \boldsymbol{y}) = \prod_{y \in \Sigma_y}\prod_{y' \in \Sigma_y} \theta_{y,y'}^{\phi_{y,y'}} \cdot \prod_{y \in \Sigma_y}\prod_{x \in \Sigma_x} \theta_{y,x}^{\phi_{y,x}} \tag{3}$$

両辺の対数をとるなら,

$$\log \Pr(\boldsymbol{x}, \boldsymbol{y}) = \sum_{y \in \Sigma_y}\sum_{y' \in \Sigma_y} \phi_{y,y'} \log \theta_{y,y'} + \sum_{y \in \Sigma_y}\sum_{x \in \Sigma_x} \phi_{y,x} \log \theta_{y,x} \tag{4}$$

ここでさらに, $\phi_{y,x}$ や $\phi_{y,y'}$ を並べたベクトルを $\Phi(\boldsymbol{x}, \boldsymbol{y})$, また $\log \theta_{y,y'}$ や $\log \theta_{y,x}$(対数をとっていることに注意) を並べたベクトルを Θ とおくと, さらに簡単に表記することができ,

$$\log \Pr(\boldsymbol{x}, \boldsymbol{y}) = \langle \Theta, \Phi(\boldsymbol{x}, \boldsymbol{y}) \rangle \tag{5}$$

となる.

さて, 話をもとに戻して, HMM を使って, 新たな $\boldsymbol{x}$ が与えられたときの $\boldsymbol{y}$ の予測を HMM で行うことを考える. そのためには, どうにかして $\Pr(\boldsymbol{x}, \boldsymbol{y})$ から条件付分布 $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を求めればよいが, ベイズの定理より,

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \frac{\Pr(\boldsymbol{x}, \boldsymbol{y})}{\Pr(\boldsymbol{x})}$$

であるから, 予測は

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in Y} \frac{\Pr(\boldsymbol{x}, \boldsymbol{y})}{\Pr(\boldsymbol{x})} \text{ あるいは } \hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in Y} \log \Pr(\boldsymbol{x}, \boldsymbol{y}) - \log \Pr(\boldsymbol{x})$$

となる. しかし実際のところ, $\Pr(\boldsymbol{x})$ は $\boldsymbol{y}$ には関係がないため, 結局,

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in Y} \log \Pr(\boldsymbol{x}, \boldsymbol{y})$$

によって予測が行えることになる.

この予測はいわゆる動的計画法により, 次の再帰式を使って効率よく求めることができる.

$$S(y_\tau) = \max_{y_1,\ldots,y_{\tau-1}} (\sum_{t=1}^{\tau} \log \theta_{y_{t-1},y_t} + \log \theta_{y_t,x_t}) = \max_{y_{\tau-1}} (\log \theta_{y_{\tau-1},y_\tau} + \log \theta_{y_\tau,x_\tau} + S(y_{\tau-1}))$$

再帰の各ステップ t において max を実現する $y_t \in \Sigma_y$ を記録しておけば, 最大値が

$$\max_{\boldsymbol{y}} \log \Pr(\boldsymbol{x}, \boldsymbol{y}) = \max_{y_T} (S(y_T) + \log \theta_{y_T,y_{T+1}})$$

によって求まったとき, 再帰を遡って, 最大値を実現する $\boldsymbol{y}$ を簡単に求めることができる.

さて, N 個の学習データが与えられたときの, モデルの推定 (すなわち, パラメータ $\theta_{y,y'}$, $\theta_{y,x}$ の推定) はどのように行えばよいだろうか. i 番目の学習データを $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ と書く, ただし $i = 1, ..., N$ とする. 通常, HMM では, この学習データをもっともよく再現するようなパラメータ, いいかえると, 学習データを生成する確率がもっとも高くなるようなパラメータにするという手法がとられる. それぞれの学習データは独立に生成されたとすると, モデルから学習データが生成される確率, すなわち尤度は,

$$\prod_{i=1}^{N} \Pr(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}; \theta)$$

となるため, これをもっとも大きくするようなパラメータ $\hat{\theta}$ が,

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \prod_{i=1}^{N} \Pr(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}; \theta)$$

によって決定される. 通常は, 対数尤度とよばれる, 確率の対数をとったものが用いられる (解は変わらない).

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \sum_{i=1}^{N} \log \Pr(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}; \theta)$$

これがいわゆる, 最尤推定とよばれる推定法である.

では, 実際に最尤推定に基づいてパラメータを求めてみよう. (3) の表現を使って, 学習データに対する対数尤度の和をつぎのようにかける.

$$\begin{aligned}
\sum_{i=1}^{N} \log \Pr(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) &= \sum_{i=1}^{N} \log \prod_{y \in \Sigma_y} \prod_{y' \in \Sigma_y} \theta_{y,y'}^{\phi_{y,y'}^{(i)}} \cdot \prod_{y \in \Sigma_y} \prod_{x \in \Sigma_x} \theta_{y,x}^{\phi_{y,x}^{(i)}} \\
&= \sum_{y \in \Sigma_y} \sum_{y' \in \Sigma_y} (\sum_{i=1}^{N} \phi_{y,y'}^{(i)}) \log \theta_{y,y'} + \sum_{y \in \Sigma_y} \sum_{x \in \Sigma_x} (\sum_{i=1}^{N} \phi_{y,x}^{(i)}) \log \theta_{y,x}
\end{aligned}$$

これを (2) の制約のもとで最大化すればよいから, ラグランジュの未定乗数法によって L は

$$L = \sum_{y \in \Sigma_y} \sum_{y' \in \Sigma_y} (\sum_{i=1}^{N} \phi_{y,y'}^{(i)}) \log \theta_{y,y'} + \sum_{y \in \Sigma_y} \sum_{x \in \Sigma_x} (\sum_{i=1}^{N} \phi_{y,x}^{(i)}) \log \theta_{y,x} - \sum_{y \in \Sigma_y} \lambda_y (\sum_{y' \in \Sigma_y} \theta_{y,y'} - 1) - \sum_{y \in \Sigma_y} \mu_y (\sum_{x \in \Sigma_x} \theta_{y,x} - 1)$$

のようにかける. ここで, λ_y, μ_y はラグランジュ定数である. これをパラメータで偏微分して 0 とおいて解けば,

$$\begin{aligned}
\frac{\partial L}{\partial \theta_{y,y'}} = \frac{\sum_{i=1}^{N} \phi_{y,y'}^{(i)}}{\theta_{y,y'}} - \lambda_y = 0 \quad &\Rightarrow \quad \theta_{y,y'} = \frac{\sum_{i=1}^{N} \phi_{y,y'}^{(i)}}{\lambda_y} \\
\frac{\partial L}{\partial \theta_{y,x}} = \frac{\sum_{i=1}^{N} \phi_{y,x}^{(i)}}{\theta_{y,x}} - \mu_y = 0 \quad &\Rightarrow \quad \theta_{y,x} = \frac{\sum_{i=1}^{N} \phi_{y,x}^{(i)}}{\mu_y}
\end{aligned}$$

ここで制約 (2) によって, λ_y と μ_y が決まり,

$$\theta_{y,y'} = \frac{\sum_{i=1}^{N} \phi_{y,y'}^{(i)}}{\sum_{y' \in \Sigma_y} (\sum_{i=1}^{N} \phi_{y,y'}^{(i)})}$$

$$\theta_{y,x} = \frac{\sum_{i=1}^{N} \phi_{y,x}^{(i)}}{\sum_{x \in \Sigma_x} (\sum_{i=1}^{N} \phi_{y,x}^{(i)})}$$

のようにパラメータが決定される.

さて, ここで 2 つの問題点が浮き上がってくる. 1 つ目は, パラメータについての制約 (2) である. 確率的な制約の意味するところは, Σ_x や Σ_y が互いに疎な事象であるということである. これはたとえば, まったく同じ位置に”man”と”woman”といった 2 つの単語が現れることがない, ということをあらわしている. これは至極当然のことではあるが, 一方, パラメータの中に, 「大文字で始まる」とか「er で終わる」とか, 単語そのものではない, 単語のもつ特徴を取り入れたいと思ったときには問題が起こってくる. たとえば, “Manchester” のように「大文字で始まる」ことと「er で終わる」ことは同時に起こりうることであり, これらはお互いに依存関係がある. これを (2) のような形で表現するのは難しい. この目的を果たすには, (2) の制約を取り払い, かつ, 確率分布として正しいモデルが望まれる.

もうひとつの問題は, 学習で用いた目的関数にある. HMM の最尤推定では, 学習データにおける入力と出力の同時確率を最大化することによって, $\Pr(\boldsymbol{x}, \boldsymbol{y})$ を精度よく求めることを目的としていた. 確かに, HMM による予測のところでもみたように, $\Pr(\boldsymbol{x}, \boldsymbol{y})$ があれば, 当然そこから $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を導くことができるため, まったく問題はなさそうだが, 果たしてこれは, $\boldsymbol{x}$ から $\boldsymbol{y}$ を予測するというわれわれの目的を「直接的に」反映しているといえるだろうか? 予測という観点からは, $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を精度よく求まれば用は足りるはずなのに, $\Pr(\boldsymbol{x}, \boldsymbol{y})$ を推定するというのは, ちょっと高望みをしているのではないだろうか? 限りあるデータから最大限に情報を引き出すという立場からは, $\Pr(\boldsymbol{y}|\boldsymbol{x})$ の精度向上を直接的に目指すべきではないだろうか?

次章で紹介する条件付確率場は, ここであげた 2 つの問に対するひとつの答えを提供するモデルである.

3 識別モデル：条件付確率場 (CRF)

この章では前章でのべた「特徴の独立性」と「条件付確率の直接推定」を実現する, いわゆる識別モデルを構成することを考える. ここでは, 識別モデルの代表格ともいえ, 現在, 自然言語処理やバイオインフォマティクスなどの分野で実用化が進んでいる条件付確率場 (Conditional Random Field; CRF) [1] を紹介する.

まず,1 つ目の「特徴の独立性」について考えてみよう. 配列の性質を表現しそうな特徴を, 独立性を無視して, とにかく列挙してみて, その集合を F としよう. これを素性集合といおう. たとえば素性のひとつ $f \in F$ としては,「y_t が名詞で, x_t は大文字で始まる」のようなものを考えられる. 同様に,「y_t が名詞で, x_t は”day”で終わる」というものも考えられる. たとえば, y_t が名詞で, x_t が”Monday”という単語だったときに上記の 2 つの特徴は両方とも成立する. これは (すくなくとも単純な)HMM では表現できなかったことである.

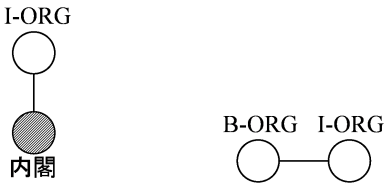
通常, CRF では素性は, 図 2 に示すように, 連続する変数の組にたいして成立する特徴として定義される. 図 3(a) の形の素性は観測素性, 図 3(b) の形の素性は遷移素性と呼ばれる. これは 2 変数の関係でモデルが記述されるという点においては HMM と同等である.

さて, それぞれの素性 $f \in F$ が, ある $(\boldsymbol{x}, \boldsymbol{y})$ の組において成立する箇所の数を $\phi_f(\boldsymbol{x}, \boldsymbol{y})$ とおき, さらに, これを並べてベクトルにしたものを $\Phi(\boldsymbol{x}, \boldsymbol{y})$ とかくことにする. それぞれの素性の重要度を θ_f とし, これをベクトルにしたものを Θ とする. これが CRF におけるパラメータである. ここで, パラメータについて HMM のような確率的な制約 (2) は存在していないことに注意する.

入力 $\boldsymbol{x}$ に対して出力 $\boldsymbol{y}$ を割り当てることの確信度合いは, パラメータベクトルと素性ベクトルの内積

$$\langle \Theta, \Phi(\boldsymbol{x}, \boldsymbol{y}) \rangle = \sum_{f \in F} \theta_f \phi_f(\boldsymbol{x}, \boldsymbol{y})$$

図 2: CRF で使用される素性



(a) (b)

によって定義する ((5) と同じ形であることに注意). しかしながら, この値は 1 より大きくなることも, マイナスになることもできるうえに, すべての $\boldsymbol{y} \in Y$ について加えても 1 にならず, このままでは確率分布になっていない. そこで, これを確率分布にするために,

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp\langle\Theta, \Phi(\boldsymbol{x}, \boldsymbol{y})\rangle}{\sum_{y \in Y} \exp\langle\Theta, \Phi(\boldsymbol{x}, \boldsymbol{y})\rangle} \tag{6}$$

のようにする. イメージとしては exp に乗せて 0 以上の値にしてから, すべての $\boldsymbol{y} \in Y$ についての和で割ることで確率分布にしている, と考えることができる. これが CRF の確率モデルである.

CRF において, 新たな $\boldsymbol{x}$ が与えられたときの $\boldsymbol{y}$ の予測を行うことを考えよう. CRF ではモデルの形自体が条件付確率分布になっているので, これをそのまま使って, 予測 $\hat{y}$ を,

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \text{ あるいは } \hat{y} = \underset{y \in Y}{\operatorname{argmax}} \log \Pr(\boldsymbol{y}|\boldsymbol{x})$$

によって求めることができる. 実際のところ, 指数関数は単調増加関数であり, (6) において, 分母は $\boldsymbol{y}$ とは関係ないため, 予測は

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \log \langle\Theta, \Phi(\boldsymbol{x}, \boldsymbol{y})\rangle$$

によって行うことができる. これは HMM における (5) と同じ形をしており, また, 素性も HMM と同じように隣り合う 2 つの変数について定義されているので, これも HMM とまったく同じ仕組みによって動的計画法で予測を行うことができる.

次に, 学習データが与えられたときの, モデルの推定を考えよう. ここで目指すべきことは「条件付確率の直接推定」をすることである. HMM と同じく, 学習データに対してもっとも高い確率を出すようにパラメータを決定する最尤推定に基づいて学習を行う. しかし, ここで異なるのは, HMM の学習のときのように $(\boldsymbol{x}, \boldsymbol{y})$ の同時確率 $\Pr(\boldsymbol{x}, \boldsymbol{y})$ を考えるのではなく, $\boldsymbol{x}$ が与えられたときの $\boldsymbol{y}$ の条件付確率 $\Pr(\boldsymbol{y}|\boldsymbol{x})$ という点である. これは, ある学習データに対して $\boldsymbol{x}^{(i)}$ が与えられたのを知っていたとするときに, $\boldsymbol{y}^{(i)}$ がもっとも高い確率で与えられるようなモデルを目指すということを意味する.

学習データにたいする (条件付の) 尤度は,

$$\prod_{i=1}^{N} \Pr(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}; \Theta)$$

となるため, これをもっとも大きくするようなパラメータ $\hat{\Theta}$ が,

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \prod_{i=1}^{N} \Pr(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}; \Theta)$$

によって決定されることになる. 対数尤度で考えれば, これは,

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log \Pr(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}; \Theta) = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log \frac{\exp \langle \Theta, \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \rangle}{\sum_{y \in Y} \exp \langle \Theta, \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \rangle}$$

となる.

では, 実際に CRF において, 学習データに対する対数尤度の和を最大化するパラメータを求めてみよう. 例によってこれをパラメータで偏微分すると,

$$\begin{aligned} \frac{\partial \sum_{i=1}^{N} \log \Pr(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}; \Theta)}{\partial \Theta} &= \sum_{i=1}^{N} \Big(\Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \frac{\sum_{y \in Y} \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \exp \langle \Theta, \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \rangle}{\sum_{y \in Y} \exp \langle \Theta, \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \rangle} \Big) \\ &= \sum_{i=1}^{N} \Big(\Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \sum_{y \in Y} \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \Pr(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}; \Theta) \Big) \end{aligned}$$

となる. 2 項目は現在のモデルにおける期待素性ベクトルともいえるもので, 動的計画法によって効率的に計算することができる. しかしながら, 偏微分は求まったが, これを 0 とおいて解いても HMM のときのように解析的に解が求められる形にはなっていない. そこで, 偏微分をもとにした, 各種数値計算法によって解を求めることになる. たとえば, 最も単純なやり方としては, 偏微分の方向が対数尤度の和を最も最大化する方向であることを利用して,

$$\Theta^{\text{新}} \leftarrow \Theta^{\text{旧}} + \eta \frac{\partial \sum_{i=1}^{N} \log \Pr(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)}; \Theta)}{\partial \Theta} \Big|_{\Theta=\Theta^{\text{旧}}}$$

のようにして, 少しずつパラメータを改善していくやり方がとられる.(η は小さな正の数)

最後に, HMM と CRF の違いについてもう一度まとめてみよう. まず, 配列の特徴を捉える表現としては, HMM では特徴の独立性の仮定が必要であったため, 素朴な HMM では用いることのできる特徴は基本的に単語レベルなど, 互いに重ならないものにしなければならないという限界があった. 一方, CRF ではこの仮定が必要ないため, よりきめ細かく, 単語よりも細かいレベルでの特徴設計ができる.

予測に関しては, HMM でも CRF でもともに動的計画法によって効率的に $\boldsymbol{y}$ を予測することができるので, 大きな違いはない. 異なるのは学習の戦略である. HMM では $\boldsymbol{x}$ と $\boldsymbol{y}$ の同時確率 $\Pr(\boldsymbol{x}, \boldsymbol{y})$, すなわち $\Pr(\boldsymbol{y}|\boldsymbol{x}) \cdot \Pr(\boldsymbol{x})$ を推定しようとしているため, $\Pr(\boldsymbol{x})$ のぶんだけ「余分な」学習をしていることとなる. 当然のことながらパラメータ空間が巨大になり, 学習データが大量でかつ均質なものでないと $\Pr(\boldsymbol{x}, \boldsymbol{y})$ の学習がうまくいかない可能性がある.

一方, CRF では $\Pr(\boldsymbol{y}|\boldsymbol{x})$ を, 各文について正規化しているため, 文単位での正解率を直接高くするように学習が行われるという違いがある. 条件付確率を直接モデル化したうえで, 識別性能 ($\boldsymbol{x}$ から $\boldsymbol{y}$ の予測性能) 改善を直接に志向した学習法をとるところが, 識別モデルと呼ばれるゆえんである.

逆にいえば, CRF は, 識別のみに特化した手法であるため, 識別以外の目的に利用することは困難である. HMM の場合 $\Pr(\boldsymbol{x}) = \sum \boldsymbol{y} \Pr(\boldsymbol{y}, \boldsymbol{x})$ とすることで言語モデルを構築することができるが, CRF の場合, $\Pr(\boldsymbol{y}|\boldsymbol{x})$ のみから $\Pr(\boldsymbol{x})$ を推定できないため, CRF を言語モデルとして使うことは原理的に難しい. 従って HMM は, 識別も言語モデリングもこなす汎用的な方法であり, CRF は識別のみに特化した手法であるといえ, 目的に合わせて使い分けるのがポイントとなろう.

4 実験による比較

4.1 実験設定

CRF の有効性を示すために日本語形態素解析の実験を行なった. 実験には京都大学テキストコーパス ver. 2.0 (**KC**) と RWCP テキストコーパス (**RWCP**) の 2 つのタグ付きコーパスを用いた. データの詳細を表 1 にまとめる.

表 1: 実験データの詳細

	KC	**RWCP**
ソース	毎日新聞 ('95)	毎日新聞 ('94)
辞書 (活用等を全展開した語彙数)	JUMAN ver. 3.61 (1,983,173)	IPADIC ver. 2.7.0 (379,010)
品詞体系のサイズ	2 階層の品詞, 活用型, 活用形, 基本形	4 階層の品詞, 活用型, 活用形, 基本形
文数 (学習)	7,958 (1 月 1 日 - 1 月 8 日の記事)	10,000 (先頭の 1 万文)
形態素数 (学習)	198,514	265,631
文数 (テスト)	1,246 (1 月 9 日の記事)	25,743 (残りすべて)
形態素数 (テスト)	31,302	655,710
素性数	791,798	580,032

CRF の 1 つの利点として, オーバラップする素性や文字種や部分文字列といった素性を素性関数という形で柔軟に投入できることにある. このような柔軟な素性設計は HMM では困難である. 表 2 にデータセット **KC** にて使用した素性関数のテンプレートをまとめる. 例えば, テンプレート $\langle bw, p1\rangle$ からは, (語彙 × 品詞) 個の素性関数が生成され, 各関数は以下のような 2 値を返す.

$$f_{1234}(\langle w', t'\rangle, \langle w, t\rangle) \stackrel{\text{def}}{=} \begin{cases} 1 & bw = \text{は} \ \& \ p1 = \text{助詞} \\ 0 & otherwise. \end{cases}$$

RWCP のテンプレートは, 品詞の階層のサイズが異なることを除けば **KC** のそれと本質的に同一である. もし着目する語が**語彙化**されている場合, つまり語の品詞が助詞, 助動詞, 接尾辞,「する, 言う」等の頻出動詞の場合は, 語彙レベルのテンプレートも用いる. このような語彙化は, 日本語形態素解析において頻繁に用いられる. 未知語処理によって生成された候補については, 語の長さ, 長さ 2 までの, 接頭/接尾辞, ひらがな/漢字/アルファベットといった文字種を用いる. また, 頻度による足切りなどは行わず, ラティス上に観察されたすべての素性を用いる. 各データでの素性数を表 1 の最下行に示す.

評価は, 以下で与えられる F 値 ($F_{\beta=1}$) で行う.

$$\begin{aligned} F_{\beta=1} &= \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}, \\ Recall &= \frac{\text{\# of correct tokens}}{\text{\# of tokens in training corpus}} \\ Precision &= \frac{\text{\# of correct tokens}}{\text{\# of tokens in system output}}. \end{aligned}$$

さらに, 正解の基準として, 以下の 3 つを設けた. 1) **seg**: 単語区切りのみ正解, 2) **top**: 単語区切りと品詞の大分類が正解, 3) **all**: 全情報が正解.

4.2 結果

表 3, 4 に **KC**, および **RWCP** の実験結果を示す. 3 つのレベルの F 値 (*seg/top/all*) を CRF と同コーパスで実験を行なった bi-gram HMM (ベースライン) についてそれぞれ示している.

KC データセットについては, ルールベースのシステム JUMAN [2]の結果も載せている. 公平な評価にするために, CRF, HMM は同じコーパスを用いて実験を行っている. **RWCP** データセットについては, ChaSen (拡張 HMM) の結果も示す. ChaSen についても CRF と同じコーパスを用いて実験を行った.

結果より, CRF は精度という点で既存手法より優れていることが分かる.

[2]JUMAN は,「未知語」という品詞を辞書に記載されていない単語に付与する. このような語は「名詞-サ変」というデフォルト品詞を与えて評価した.

表 2: 素性テンプレート: $f_k(\langle w', t' \rangle, \langle w, t \rangle)$
$t' = \langle p1', p2', cf', ct, bw' \rangle$, $t = \langle p1, p2, cf, ct, bw \rangle$,
ただし $p1'/p1$, $p2'/p2$, cf'/cf, ct'/ct bw'/bw は, それぞれ 語 w'/w の品詞大分類, 再分類, 活用型, 活用形, 基本形である.

タイプ	テンプレート
uni-gram 基本素性	$\langle p1 \rangle$ $\langle p1, p2 \rangle$
w 既知	$\langle bw \rangle$ $\langle bw, p1 \rangle$ $\langle bw, p1, p2 \rangle$
w 未知	w の文字列長 サイズ 2 までの接尾 $\times \{\phi, \langle p1 \rangle, \langle p1, p2 \rangle\}$ サイズ 2 までの接頭 $\times \{\phi, \langle p1 \rangle, \langle p1, p2 \rangle\}$ 文字種 $\times \{\phi, \langle p1 \rangle, \langle p1, p2 \rangle\}$
bi-gram 基本素性	$\langle p1', p1 \rangle$ $\langle p1', p1, p2 \rangle$ $\langle p1', p2', p1 \rangle$ $\langle p1', p2', p1, p2 \rangle$ $\langle p1', p2', cf', p1, p2 \rangle$ $\langle p1', p2', ct', p1, p2 \rangle$ $\langle p1', p2', cf', ct', p1, p2 \rangle$ $\langle p1', p2', p1, p2, cf \rangle$ $\langle p1', p2', p1, p2, ct \rangle$ $\langle p1', p2', p1, p2, cf, ct \rangle$ $\langle p1', p2', cf', p1, p2, cf \rangle$ $\langle p1', p2', ct, p1, p2, ct \rangle$ $\langle p1', p2', cf', p1, p2, ct \rangle$ $\langle p1', p2', ct', p1, p2, cf \rangle$ $\langle p1', p2', cf', ct', p1, p2, cf, ct \rangle$
w' 語彙化	$\langle p1', p2', cf', ct', bw', p1, p2 \rangle$ $\langle p1', p2', cf', ct', bw', p1, p2, cf \rangle$ $\langle p1', p2', cf', ct', bw', p1, p2, ct \rangle$ $\langle p1', p2', cf', ct', bw', p1, p2, cf, ct \rangle$
w 語彙化	$\langle p1', p2', p1, p2, cf, ct, bw \rangle$ $\langle p1', p2', cf', p1, p2, cf, ct, bw \rangle$ $\langle p1', p2', ct', p1, p2, cf, ct, bw \rangle$ $\langle p1', p2', cf', ct', p1, p2, cf, ct, bw \rangle$
w'/w 共に語彙化	$\langle p1', p2', cf', ct', bw', p1, p2, cf, ct, bw \rangle$

4.3 コーパスサイズと精度の関係

CRF は条件付き確率 $\Pr(\boldsymbol{y}|\boldsymbol{x})$ の最尤推定であり, 文単位での正解率を直接高くするような学習が行われる. 一方, HMM は, 同時確率 $\Pr(\boldsymbol{y}, \boldsymbol{x}) = \Pr(\boldsymbol{y}|\boldsymbol{x}) \cdot \Pr(\boldsymbol{x})$ の最尤推定であるため, 学習時に言語モデル $\Pr(\boldsymbol{x})$ の推定を同時に行っていることになる. 言語モデルの推定は一般に大量の文書が必要であり, 少量のデータでは推定が不安定になり, 高い精度を期待できない.

この事実を検証するために, HMM と CRF について学習データのサイズと精度の関係を調べた. 実験には **RWCP** コーパスを用いた. 図 5 に学習文数と F 値 (All) の関係を示す. 図から明らかなように CRF は学習曲線の立ちあがりが良好である. おおむね CRF は HMM の数〜数十分の一のコーパスで同程度の性能が達成できることが分かる.

5 CRF が利用可能なツール

世の中には, 本稿で取り上げた CRF を実際に使うことのできるツールがすでにいくつか公開されている.

5.1 MALLET: A Machine Learning for Language Toolkit

http://mallet.cs.umass.edu/index.php/Main_Page

MALLET は Andrew McCallum 氏が中心となって開発した言語処理向けの機械学習ツールキットである. Java で実装されており,CRF の他に, 文書分類, クラスタリング, 情報抽出といったツールが同封されている.

5.2 CRF Project Page

http://crf.sourceforge.net/

Sunita Sarawagi 氏が開発, 公開している Java で実装されたツールである. API の提供がメインであり, 単体

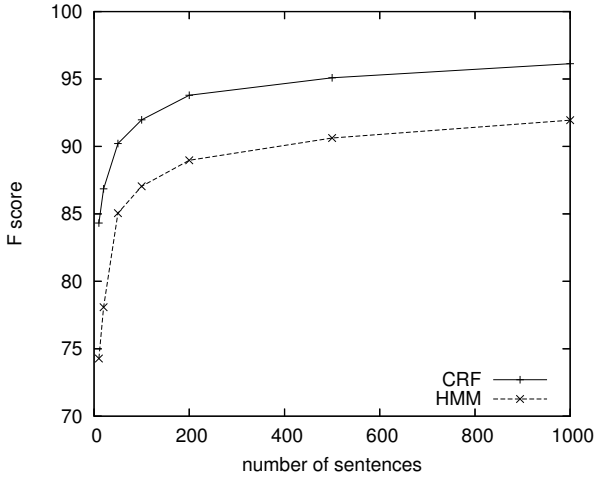
図 3: 実験結果: KC

system	$F_{\beta=1}$ (*seg* / *top* / *all*))
CRF	98.96 / 98.31 / 96.75
HMM	96.22 / 94.99 / 91.85
JUMAN	98.70 / 98.09 / 94.35

図 4: 実験結果: RWCP

system	$F_{\beta=1}$ (*seg* / *top* / *all*))
CRF	99.11 / 98.72 / 97.65
HMM	96.42 / 95.81 / 94.16
ChaSen	98.86 / 98.38 / 97.00

図 5: 学習データ量と精度の関係



での使用よりはむしろ, 他のプログラムに組み込みを意識した設計となっている.

5.3 FlexCRF

http://www.jaist.ac.jp/ hieuxuan/flexcrfs/flexcrfs.html
北陸先端大の Xuan-Hieu Phan, Le-Minh Nguyen 両氏が開発, 公開しているツールキットである. 単体での動作を前提としているため, CRF の実験には適切である. また, 1st-order の他に 2nd-order (trigram) の CRF をサポートしている. さらに並列計算機を使って大量データの学習を行うツールも公開されている.

5.4 CRF++

http://chasen.org/ taku/software/CRF++/
著者の 1 人, 工藤が公開しているオープンソースの CRF ツールキットである. 簡単に CRF を導入できるようコンパクトな設計となっている. FlexCRF に比べると多機能ではないが, 基本的な機能はしっかり押さえられている. 特筆すべき点として, 学習/解析ともに他のツールに比べ高速に動作する. さらに,N-best 解の出力や周辺確率の計算といった機能を持っている.

5.5 MeCab

http://mecab.sourceforge.jp/
MeCab は CRF を採用した初めての形態素解析器である. IPA 品詞体系の辞書と JUMAN 品詞体系の辞書を標準でサポートする. パラメータ推定パッケージが含まれているため, 基本的にコーパスと辞書さえあれば独自の形態素解析器を作成することが可能となっている. 今後 IPA, JUMAN 以外の辞書をサポートする予定である.

参考文献

[1] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[2] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 1999.

[3] E. F. Tjong and K. Sang. Text chunking by system combination. In *Proceedings of Conference on Computational Natural Language Learning*, 2000.