可変次数 Linear-Chain CRF の効率的な計算法

真鍋宏史
京都大学 情報学研究科
manabe@nlp.kuee.kyoto-u.ac.jp Twitter ID:@takeda25

1　研究の概要

系列ラベリング問題に対する手法の一つに，Conditional Random Fields(CRF)[1] というものがある．これは系列全体に最大エントロピーモデルを適用するというものであり，観測とラベル間の柔軟な関係性を記述できる．CRF の中でよく使われるものは Linear-Chain CRF と呼ばれるもので，一般には一次のマルコフ性を仮定し，前向き・後ろ向きアルゴリズムという一種の動的計画法によって素性の期待値を効率的に計算する．しかし，この手法では次数に対して指数的にノード数が増加し，計算量もそれに比例するため，高次素性の利用が現実的に難しい．Ye ら [4] の研究では，高次の素性を導入しつつ多項式時間に計算量を収める手法を紹介し，高次素性の使用が様々なタスクにおいて有効であることを示しているが，本研究では，より単純かつ効率的に期待値を計算する手法を提案する．本手法においては，「前向き差分得点」を求めるときに「前向き合計得点」を，「後ろ向き合計得点」を求めるときに「後ろ向き差分得点」をそれぞれ補助的に使用することで，動的計画法の適用を可能にしている．

2　関連研究

Ye ら [4] は，本研究と同じく，多項式時間で高次 Linear-Chain CRF のパラメータ推定を行う方法を提案しているが，彼らの手法においては，素性関数のユニークなラベル列数を M，素性関数の最大次数を K，訓練データの長さを X とするとき，悲観的な見積もりでは一回のイテレーションに $O(M^3K^4X)$ の計算量が必要となる．本研究での計算複雑性は，訓練データ中のそれぞれの位置でアクティブになる素性関数が持つユニークなラベル列の長さの合計であり，これは最大でも $O(MKX)$ である．

3　可変次数 Linear-Chain CRF

本研究では，Ye ら [4] と同じく，Linear-Chain CRF において素性関数ごとに異なる Markov 次数をを設定するモデルを考える．ここでは，それを「可変次数 Linear-Chain CRF」と呼ぶ．

3.1　素性関数の生成

一次の Linear-Chain CRF において，素性関数の生成にあたって，一般的に使用者は観測についての関数を設定し，それとラベル遷移と組み合わせ，プログラムが実際の素性関数を生成する．このとき，可能なラベル遷移をすべて列挙する方法と，次に述べる「疎素性集合」の方法がある．

3.1.1　疎素性集合

使用者が観測についての関数を記述し，プログラムがそれによって素性関数を生成するとき，すべての可能な組み合わせを生成する代わりに，訓練データ中で出現する組み合わせのみを生成するという方法がある．この機能を実装している CRFSuite[2] による実験では，大きく精度を損なうことなく，素性数やモデルの大きさを減らすことができている．

3.1.2　可変次数 Linear-Chain CRF での素性関数の生成

可変次数 Linear-Chain CRF においても，素性関数の生成にあたっては，まず使用者が観測に関する関数，あるいはそれぞれの位置におけるそれらの値を設定する．本モデルでは，疎素性集合の考え方により，それらが真となる訓練データ中での位置と，そこまでのラベル列とを組み合わせて，任意の次数の素性関数を生成する．

素性関数を $f_1, \cdots, f_m$ とする．素性関数 f_i は，二値関数 $b_i(\mathbf{x}, t)$ と，二値関数 $L_i(\mathbf{y}, t)$ の積として表されるとする．

$$f_i(\mathbf{x}, \mathbf{y}, t) = b_i(\mathbf{x}, t)L_i(\mathbf{y}, t) \tag{1}$$

ここで，$L_i(\mathbf{y}, t)$ は位置 t までのラベル列の関数である．

3.2　確率分布

CRF では，確率分布は

$$P(\mathbf{y} \mid \mathbf{x}) = Z_{\mathbf{x}}(\mathbf{y})/Z_{\mathbf{x}} \tag{2}$$

として表される．

$$Z_{\mathbf{x}} = \sum_{\mathbf{y}} Z_{\mathbf{x}}(\mathbf{y}) \tag{3}$$

である．可変次数 Linear-Chain CRF では，λ_i を素性関数の重みとすると，

$$Z_{\mathbf{x}}(\mathbf{y}) = \exp\left(\sum_{i=1}^{m}\sum_{t=1}^{T+1} \lambda_i f_i(\mathbf{x}, \mathbf{y}, t)\right) \tag{4}$$

となる．($T+1$ は系列終端位置を表す)

3.3　パラメータ推定

訓練データ $\mathcal{T}$ が与えられたとき，モデルのパラメータ λ_i は，正規化された対数尤度

$$\mathcal{L}_{\mathcal{T}} = \log \prod_{(\mathbf{x}, \tilde{\mathbf{y}}) \in \mathcal{T}} P(\tilde{\mathbf{y}} \mid \mathbf{x}) - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma_{reg}^2} \tag{5}$$

を最大化するように選ぶ．σ_{reg} は正規化のためのパラメータである．$\mathcal{L}_{\mathcal{T}}$ は $\lambda_1, \cdots, \lambda_i$ について凹関数であるため，

$$\frac{\partial \mathcal{L}_{\mathcal{T}}}{\partial \lambda_i} = \tilde{E}(f_i) - E(f_i) - \frac{\lambda_i}{\sigma_{reg}^2} = 0 \tag{6}$$

のときに最大値をとる．可変次数 Linear-Chain CRF において，$\tilde{E}(f_i) = \sum_{(\mathbf{x},\tilde{\mathbf{y}})\in\mathcal{T}} \sum_{t=1}^{|\mathbf{x}|+1} f_i(\mathbf{x},\tilde{\mathbf{y}},t)$ は素性関数 f_i の訓練データ中での経験的な出現回数であり，$E(f_i) = \sum_{(\mathbf{x},\tilde{\mathbf{y}})\in\mathcal{T}} \sum_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}) \sum_{t=1}^{|\mathbf{x}|+1} f_i(\mathbf{x},\mathbf{y},t)$ は f_i の出現回数の期待値である．$E(f_i)$ を求めるにあたって，本研究では次章で述べる「合計・差分アルゴリズム」を使用する．その後，このアルゴリズムによって得られる勾配と，$\mathcal{L}_{\mathcal{T}}$ の値によって，L-BFGS 法により対数尤度を最大化する．

4 可変次数 Linear-Chain CRF のアルゴリズム

可変次数 Linear-Chain CRF において，パラメータ推定とデコードのためにそれぞれ計算法が必要となるが，本論文ではパラメータ推定のための計算法を提案する．

4.1 合計・差分アルゴリズム

ここでは，素性関数 f_i の期待値 $E(f_i) = \sum_{(\mathbf{x},\tilde{\mathbf{y}})\in\mathcal{T}} \sum_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}) \sum_{t=1}^{|\mathbf{x}|+1} f_i(\mathbf{x},\mathbf{y},t)$ を求めるために，「合計・差分アルゴリズム」という名前の以下に述べる計算法を提案する．

4.1.1 表記

$\mathbf{x},\mathbf{y}$ はそれぞれ長さ T の観測列・ラベル列を表す．$|\cdot|$ は列の長さを表す．ラベルの集合は $\mathcal{Y} = \{l_1,\cdots,l_L\}$ とする．また，位置 0 と位置 $T+1$ を仮想的に考え，それぞれの位置でのラベルを l_{BOS}, l_{EOS} とする．

位置 t でのラベル集合 $\mathcal{Y}_t$ を次のように定義する．

$$\mathcal{Y}_t = \begin{cases} \{l_{BOS}\} & \text{if } t=0 \\ \{l_{EOS}\} & \text{if } t=T+1 \\ \mathcal{Y} & \text{otherwise} \end{cases} \tag{7}$$

ラベル集合の列 $\mathcal{Y}_n,\cdots,\mathcal{Y}_m$ に対して，あるラベル列 $\mathbf{z}$ が $|\mathbf{z}| = m-n+1, \forall(t: 1\le t\le m-n+1)\mathbf{z}_t \in \mathcal{Y}_{n+t-1}$ を満たすとき，$\mathbf{z}\in\mathcal{Y}_{n:m}$ とする．

任意のラベル列 $\mathbf{z}$ に対し，$\mathbf{z}_{i:j} \stackrel{\text{def}}{=} \mathbf{z}_{i\cdots j}$ とする．$j<i$ であれば，$\mathbf{z}_{i:j}$ は空の列 (ϵ と表す) とする．また，表記の都合上，任意のラベル列について述べるとき，$\mathbf{z}^1,\mathbf{z}^2,\cdots$ のように右上に通し番号を付ける．

あるラベル列 $\mathbf{z}^1$ とラベル列 $\mathbf{z}^2$ をこの順に連結したものが $\mathbf{z}^3$ となるとき，$\mathbf{z}^3 = \mathbf{z}^1 + \mathbf{z}^2$ とする．

あるラベル列 $\mathbf{z}^1$ がラベル列 $\mathbf{z}^2$ の接尾辞である（$\mathbf{z}^2$ が $\mathbf{z}^1$ を含み，末尾を共有する）とき，$\mathbf{z}^1 \le_s \mathbf{z}^2$ と表記する ($\mathbf{z}^1 = \mathbf{z}^2$ を含む)．任意のラベル列 $\mathbf{z}$ に対し，$\epsilon \le_s \mathbf{z}$ とする．また，$\mathbf{z}^1$ が $\mathbf{z}^2$ の接尾辞であり，$\mathbf{z}^1 \ne \mathbf{z}^2$ であるとき，$\mathbf{z}^1 <_s \mathbf{z}^2$ と表記する．このとき，次の式が明らかに成り立つ．

$$\mathbf{z}^1 <_s \mathbf{z}^2, \mathbf{z}^1 \ne \epsilon \Rightarrow \mathbf{z}^1_{1:|\mathbf{z}^1|-1} <_s \mathbf{z}^2_{1:|\mathbf{z}^2|-1} \tag{8}$$

また，次も明らかに成り立つ．

$$\mathbf{z}^1 <_s \mathbf{z}^3, \mathbf{z}^2 <_s \mathbf{z}^3 \Rightarrow \mathbf{z}^1 <_s \mathbf{z}^2 \ \text{ or } \ \mathbf{z}^2 \le_s \mathbf{z}^1 \tag{9}$$

また，$\mathbf{z}^2$ が，集合 $\mathcal{S}$ に含まれる要素の中で最も長い $\mathbf{z}^1$ の接尾辞 ($\mathbf{z}^1$ 自身を除く) であるとき，それを $s(\mathbf{z}^1,\mathcal{S}) = \mathbf{z}^2$ と表わし，$\mathbf{z}^2$ を $\mathcal{S}$ に関する $\mathbf{z}^1$ の最長接尾辞であるとする．定義は次のようになる．

$$\begin{aligned} s(\mathbf{z}^1,\mathcal{S}) = \mathbf{z}^2 \ \text{ if and only if } \quad & \mathbf{z}^2 \in \mathcal{S} \ \text{ and } \ \mathbf{z}^2 <_s \mathbf{z}^1 \\ \text{and } \ \forall(\mathbf{z}\in\mathcal{S})\mathbf{z} <_s \mathbf{z}^1 \Rightarrow \mathbf{z} \le_s \mathbf{z}^2 & \end{aligned} \tag{10}$$

なお，任意の $\mathcal{S}$ に対し，$s(\epsilon,\mathcal{S}) = \perp$ とおく．$\perp$ は実体を持たない仮想的なラベル列で，他のラベル列と比較したときに一致することはない（$\forall(\mathbf{z})\mathbf{z} \ne \perp$ である）．

また，$\perp^n$ を，他のラベル列と比較したときに一致することのない，長さ n のラベル列とする ($\forall(\mathbf{z})\mathbf{z} \ne \perp^n, |\perp^n| = n$ である)．

4.1.2 定義

素性関数 f_i は，式 (1) のように関数 b_i, L_i の積として表される．L_i は対応するラベル列 $\mathbf{z}^i$ を持っており，次のように定義する．

$$L_i(\mathbf{y},t) = \begin{cases} 1 & \text{if } \mathbf{y}_{t-|\mathbf{z}^i|+1:t} = \mathbf{z}^i \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$\mathbf{z}^i$ と関連づけられた L_i を構成要素に持つ f_i を，次数 $|\mathbf{z}^i|-1$ の素性関数と呼ぶ．

位置 t での「パス集合」を次のように定義する．

$$\mathcal{P}_t = \mathcal{Y}_t \cup \{\epsilon\} \cup \bigcup_{t'=t}^{T} \{\mathbf{z}^k_{1:|\mathbf{z}^k|-(t'-t)} \mid b_k(\mathbf{x},t') = 1, \mathbf{z}^k \in \mathcal{Y}_{t'-|\mathbf{z}^k|+1:t'}\} \tag{12}$$

$\mathcal{P}_t$ を位置 t における「パス集合」とし，その要素を位置 t における「パス」とする．パス集合をこのように定義することで，次が成り立つ．

$$\mathbf{z} \in \mathcal{P}_t, \mathbf{z} \ne \epsilon, t > 0 \Rightarrow \mathbf{z}_{1:|\mathbf{z}|-1} \in \mathcal{P}_{t-1} \tag{13}$$

あるパス $\mathbf{z}^p \in \mathcal{P}_t$ と位置 $t \ge 1$ について，$w(\mathbf{z}^p,t) \stackrel{\text{def}}{=} \sum_{i:\mathbf{z}^i=\mathbf{z}^p, b_i(\mathbf{x},t)=1} \lambda_i$ とする．また，$W(\mathbf{z}^p,t) \stackrel{\text{def}}{=} \sum_{i:\mathbf{z}^i\in\mathcal{P}_t, \mathbf{z}^i \le_s \mathbf{z}^p} w(\mathbf{z}^i,t)$ とする．このように定義すると，任意の $\mathbf{z}^p \ne \epsilon$ について，$W(\mathbf{z}^p,t) = W(s(\mathbf{z},\mathcal{P}_t),t) + w(\mathbf{z}^p,t)$ がいえる．

あるパス $\mathbf{z}^p \ne \epsilon \in \mathcal{P}_t$ と位置 $t \ge 1$ について，「前向き差分得点」である $\alpha(\mathbf{z}^p,t)$ を次のように定義する．

$$\begin{aligned} \alpha(\mathbf{z}^p,t) \stackrel{\text{def}}{=} & \sum_{\mathbf{z}:\mathbf{z}\in\mathcal{Y}_{0:t}, \mathbf{z}^p \le_s \mathbf{z}} \exp\left(\sum_{t'=1}^{t-1} \sum_{i:f(\mathbf{x},\mathbf{z},t')=1} \lambda_i\right) - \\ & \sum_{\mathbf{z}:\mathbf{z}\in\mathcal{Y}_{0:t}, \exists(\mathbf{z}'\in\mathcal{P}_t, s(\mathbf{z}',\mathcal{P}_t)=\mathbf{z}^p)\mathbf{z}' \le_s \mathbf{z}} \\ & \exp\left(\sum_{t'=1}^{t-1} \sum_{i:f(\mathbf{x},\mathbf{z},t')=1} \lambda_i\right) \end{aligned} \tag{14}$$

これは，$\mathbf{z}^p$ を接尾辞として含む系列の位置 1 から $t-1$ までの前向き得点の合計から，「$\mathbf{z}^p$ を最長接尾辞とする位置 t のパス」のいずれかを接尾辞として含む系列の位置 1 から $t-1$ までの前向き得点の合計を引いた差分である．$\alpha(\epsilon,t) = 0$ とする．

また，あるパス $\mathbf{z}^p \in \mathcal{P}_t$ と位置 t について，「前向き合計得点」である $\gamma(\mathbf{z}^p,t)$ を次のように定義する．

$$\gamma(\mathbf{z}^p,t) \stackrel{\text{def}}{=} \sum_{\mathbf{z}:\mathbf{z}\in\mathcal{Y}_{0:t}, \mathbf{z}^p \le_s \mathbf{z}} \exp\left(\sum_{t'=1}^{t} \sum_{i:f(\mathbf{x},\mathbf{z},t')=1} \lambda_i\right) \tag{15}$$

これは，$\mathbf{z}^p$ を接尾辞として含む，位置 t までの系列の前向き得点を合計したものである．$\gamma(\epsilon,0) = \gamma(l_{BOS},0) = 1$ とする．

このように定義すると，$t \geq 1$ で次の等式が成り立つ．

$$\alpha(\mathbf{z}^p, t) = \gamma(\mathbf{z}^p_{1:|\mathbf{z}^p|-1}, t-1) - \sum_{\mathbf{z}:\mathbf{z}\in\mathcal{P}_t, s(\mathbf{z},\mathcal{P}_t)=\mathbf{z}^p} \gamma(\mathbf{z}_{1:|\mathbf{z}|-1}, t-1) \quad (16)$$

$$\gamma(\mathbf{z}^p, t) = \sum_{\mathbf{z}\in\mathcal{P}_t, \mathbf{z}^p \leq_s \mathbf{z}} \alpha(\mathbf{z}, t)\exp(W(\mathbf{z}, t)) \quad (17)$$

また，あるパス $\mathbf{z}^p \in \mathcal{P}_t$ と位置 t について，「後ろ向き合計得点」である $\beta(\mathbf{z}^p, t)$ を次のように定義する．

$$\beta(\mathbf{z}^p, t) \stackrel{\mathrm{def}}{=} \sum_{\mathbf{z}\in\mathcal{Y}_{t+1:T+1}} \exp\left(\sum_{t'=t+1}^{T+1} \sum_{i:f_i(\mathbf{x},\mathbf{z}^p+\mathbf{z},t')} \lambda_i\right) \quad (18)$$

これは，位置 $t+1$ から $T+1$ までのすべての系列について，それが $\mathbf{z}^p$ を接頭辞として持つときの後ろ向き得点を合計したものである．任意の $\mathbf{z}^p \in \mathcal{P}_{t+1}$ について，$\beta(\mathbf{z}^p, T+1) = 1$ とする．なお，$f_i(\mathbf{x}, \mathbf{z}^p + \mathbf{z}, t')$ は正確には $f_i(\mathbf{x}, \perp^{t+1-|\mathbf{z}^p|} + \mathbf{z}^p + \mathbf{z}, t')$ であるが，このように略記する．

さらに，あるパス $\mathbf{z}^p \in \mathcal{P}_t$ と位置 $t \leq T$ について，$\delta(\mathbf{z}^p, t)$ を次のように定義する．

$$\delta(\mathbf{z}^p, t) \stackrel{\mathrm{def}}{=} \sum_{\mathbf{z}\in\mathcal{Y}_{t+1:T+1}} \left(\exp\left(\sum_{t'=t+1}^{T+1} \sum_{i:f_i(\mathbf{x},\mathbf{z}^p+\mathbf{z},t')=1} \lambda_i\right) - \exp\left(\sum_{t'=t+1}^{T} \sum_{i:f_i(\mathbf{x},s(\mathbf{z}^p,\mathcal{P}_t)+\mathbf{z},t')=1} \lambda_i\right)\right) \quad (19)$$

これは，位置 $t+1$ から T までの系列について，それが $\mathbf{z}^p$ を接頭辞を持つとしたときの後ろ向き得点から，「$\mathbf{z}^p$ の最長接尾辞」を接頭辞と持つとしたときの後ろ向き得点を引いた差分の合計である．

このように定義すると，$t \leq T$ で次の等式が成り立つ．

$$\delta(\mathbf{z}^p, t) = \sum_{\mathbf{z}\in\mathcal{P}_{t+1}, \mathbf{z}_{1:|\mathbf{z}|-1}=\mathbf{z}^p} (\beta(\mathbf{z}, t+1)\exp(W(\mathbf{z}, t+1)) - \beta(s(\mathbf{z}, \mathcal{P}_{t+1}), t+1)\exp(W(s(\mathbf{z}, \mathcal{P}_{t+1}), t+1)) \quad (20)$$

$$\beta(\mathbf{z}^p, t) = \sum_{\mathbf{z}\in\mathcal{P}_t, \mathbf{z}\leq_s \mathbf{z}^p} \delta(\mathbf{z}, t) \quad (21)$$

式 (20) の導出はそれほど自明ではないため，詳しく述べる．まず，式 (19) を次のように変形する．

$$\sum_{\mathbf{z}\in\mathcal{Y}_{t+2:T+1}} \left(\sum_{\mathbf{z}'\in\mathcal{Y}_{t+1:t+1}} \left(\exp\left(\sum_{t'=t+2}^{T+1} \sum_{i:f_i(\mathbf{x},\mathbf{z}^p+\mathbf{z}'+\mathbf{z},t')=1} \lambda_i\right)\exp(W(\mathbf{z}, t+1)) - \exp\left(\sum_{t'=t+2}^{T} \sum_{i:f_i(\mathbf{x},s(\mathbf{z}^p,\mathcal{P}_t)+\mathbf{z}'+\mathbf{z},t')=1} \lambda_i\right) \exp(W(s(\mathbf{z}, \mathcal{P}_{t+1}), t+1))\right)\right) \quad (22)$$

このとき，$\mathbf{z}' \in \mathcal{Y}_{t+1:t+1}$ のうち，$\mathbf{z}^p + \mathbf{z}' \notin \mathcal{P}_{t+1}$ のものについては，括弧の中が 0 となるために省略できる．もしそうでないとすれば，$s(\mathbf{z}^p, \mathcal{P}_t) + \mathbf{z}' <_s \mathbf{z}'' <_s \mathbf{z}^p + \mathbf{z}', \mathbf{z}'' \in \mathcal{P}_{t+1}$ となるようなラベル列 $\mathbf{z}''$ が存在し, 式 (13) より $\mathbf{z}''_{1:|\mathbf{z}''|-1} \in \mathcal{P}_t$ となる．また，式 (8) より $s(\mathbf{z}^p, \mathcal{P}_t) <_s \mathbf{z}''_{1:|\mathbf{z}''|-1} <_s \mathbf{z}^p$ となる．これは，式 (10) の最長接尾辞の定義に矛盾する．

よって，$\mathbf{z}^p + \mathbf{z}' \in \mathcal{P}_{t+1}$ の場合だけを考慮すればよい．このとき，$s(\mathbf{z}^p + \mathbf{z}', \mathcal{P}_{t+1}) \leq_s s(\mathbf{z}^p, \mathcal{P}_t) + \mathbf{z}'$ である．もしそうでないとすれば，$s(\mathbf{z}^p + \mathbf{z}', \mathcal{P}_{t+1}) <_s \mathbf{z}^p + \mathbf{z}', s(\mathbf{z}^p, \mathcal{P}_t) + \mathbf{z}' <_s \mathbf{z}^p + \mathbf{z}'$ であるため，式 (9) より，$s(\mathbf{z}^p, \mathcal{P}_t) + \mathbf{z}' <_s s(\mathbf{z}^p + \mathbf{z}', \mathcal{P}_{t+1})$ となる．$\mathbf{z}'' = s(\mathbf{z}^p + \mathbf{z}', \mathcal{P}_{t+1})$ とおくと，$\mathbf{z}'' \in \mathcal{P}_{t+1}$ であり，$\mathbf{z}'' \neq \epsilon$ であるため，式 (13) より $\mathbf{z}''_{1:|\mathbf{z}''|-1} \in \mathcal{P}_t$ となり，また式 (8) より，$s(\mathbf{z}^p, \mathcal{P}_t) <_s \mathbf{z}''_{1:|\mathbf{z}''|-1} <_s \mathbf{z}^p$ となるが，これは式 (10) の最長接尾辞の定義に矛盾する．また，同じく式 (10) より，位置 $t+1$ に $s(\mathbf{z}^p + \mathbf{z}', \mathcal{P}_{t+1}) <_s \mathbf{z}'' <_s \mathbf{z}^p + \mathbf{z}'$ であるような $\mathbf{z}''$ は存在しない．そのため，式 (22) は次のように書き直せる．

$$\sum_{\mathbf{z}\in\mathcal{Y}_{t+2:T+1}} \left(\sum_{\mathbf{z}'\in\mathcal{P}_{t+1}, \mathbf{z}'_{1:|\mathbf{z}'|-1}=\mathbf{z}^p} \left(\exp\left(\sum_{t'=t+2}^{T+1} \sum_{i:f_i(\mathbf{x},\mathbf{z}'+\mathbf{z},t')=1} \lambda_i\right) - \exp\left(\sum_{t'=t+2}^{T} \sum_{i:f_i(\mathbf{x},s(\mathbf{z}',\mathcal{P}_{t+1})+\mathbf{z},t')=1} \lambda_i\right)\right)\right) \quad (23)$$

ここから，式 (20) が導ける．

また, あるパス $\mathbf{z}^p \in \mathcal{P}_t$ と位置 t について, $\theta(\mathbf{z}^p, t), \sigma(\mathbf{z}^p, t)$ を次のように定義する．

$$\theta(\mathbf{z}^p, t) \stackrel{\mathrm{def}}{=} \alpha(\mathbf{z}^p, t)\exp(W(\mathbf{z}^p, t))\beta(\mathbf{z}^p, t) \quad (24)$$

$$\sigma(\mathbf{z}^p, t) \stackrel{\mathrm{def}}{=} \sum_{\mathbf{z}:\mathbf{z}\in\mathcal{P}_t, \mathbf{z}\leq_s \mathbf{z}^p} \theta(\mathbf{z}, t) \quad (25)$$

$\theta(\mathbf{z}^p, t)$ は，位置 t までの部分系列が $\mathbf{z}^p$ を含み，かつ「$\mathbf{z}^p$ を接尾辞として含む，位置 t のパス」のいずれも接尾辞として含まないような系列の $t = 0$ から $T+1$ までの得点を合計したものである．$\sigma(\mathbf{z}^p, t)$ は，位置 t までの部分系列が $\mathbf{z}^p$ を含むような系列の $t = 0$ から $T+1$ までの得点を合計したものである．$\sigma(\mathbf{z}^p, t) = \sigma(s(\mathbf{z}^p, \mathcal{P}_t), t) + \theta(\mathbf{z}^p, t)$ となる．

正規化係数 $Z_{\mathbf{x}}$ は，γ を使って次のように表せる．

$$Z_{\mathbf{x}} = \gamma(\epsilon, T+1) \quad (26)$$

ラベル列 $\mathbf{y}$ の中で，位置 t で終わる場所でパス $\mathbf{z}^p \in \mathcal{P}_t$ が現れる確率は次のように表せる．

$$P(\mathbf{z}^p \leq_s \mathbf{y}_{1:t} \mid \mathbf{x}) = \sigma(\mathbf{z}^p, t)/Z_{\mathbf{x}} \quad (27)$$

4.1.3 手続き

ここでは，前節で定義した値を効率的に求める手続きを示す．

まず，$t = 1, \cdots, T+1$ について，$\mathcal{P}_t$ を求める．また，位置 t でラベル列 $\mathbf{z}$ を持つパスに対応する素性関数の集合を $\mathcal{F}_{\mathbf{z},t}$ とし，それらを列挙する．この手続きは，Algorithm 1 で表せる．任意の $w(\mathbf{z}, t)$，また他のすべての数値変数は 0 で初期化されているものとする．この手続きの中では明示的に記述していないが，この過程で，位置 t でのあるパス $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ に対して，それに対応する位置 $t-1$ のパス $\mathbf{z}_{1:|\mathbf{z}|-1} \in \mathcal{P}_{t-1}$（以下,「接頭辞パス」と呼ぶ）を容易に関連付けることができる．また，パス集合を，パスのラベル列を逆に並べたものをキーとするトライによって管理することによって，位置 t でのある $\mathbf{z}$ に対して，それに対応する最長

Algorithm 1 Make paths

for $t = 1$ **to** $T+1$ **do**
  $\mathcal{F} \leftarrow \{f_k \mid b_k(\mathbf{x}, t) = 1, \mathbf{z}^k \in \mathcal{Y}_{t-|\mathbf{z}^k|+1:t}\}$
  for all $f_i \in \mathcal{F}$ **do**
    $\mathcal{F}_{\mathbf{z}^i,t} \leftarrow \mathcal{F}_{\mathbf{z}^i,t} \cup \{f_i\}$
    for $j = 0$ **to** $|\mathbf{z}^i|$ **do**
      $\mathcal{P}_{t-j} \leftarrow \mathcal{P}_{t-j} \cup \{\mathbf{z}^i_{1:|\mathbf{z}|-j}\}$
    end for
  end for
end for

接尾辞 $s(\mathbf{z}, P_t)$ を関連付けることができる．また，トライを深さ優先で探索することにより，$\mathbf{z}^1 <_s \mathbf{z}^2$ のときに $\mathbf{z}^1$ が $\mathbf{z}^2$ よりも前になるような順序で並べることができる．この順序を「昇順」(ascending order) と定義し，その逆を「降順」(descending order) と定義する．これらの情報（パスに対応する素性関数の集合，接頭辞パス，最長接尾辞パス，パスの並び順）はイテレーションにかかわらず不変であるため，記録しておくことにより重複計算を避けられる．

Algorithm 2 は，イテレーションごとに実行するものである．この手順によって，例えば，式 (16) が満たされること

Algorithm 2 Sum-difference

1: **for** $t = 1$ **to** $T+1$ **do**
2:   **for all** $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ in ascending order **do**
3:     **for all** $f_i \in \mathcal{F}_{\mathbf{z},t}$ **do**
4:       $w(\mathbf{z}, t) \leftarrow w(\mathbf{z}, t) + \lambda_i$
5:     **end for**
6:     $W(\mathbf{z}, t) \leftarrow W(s(\mathbf{z}, \mathcal{P}_t), t) + w(\mathbf{z}, t)$
7:   **end for**
8: **end for**
9: $\gamma(\epsilon, 0) \leftarrow 1, \gamma(l_{BOS}, 0) \leftarrow 1$
10: **for** $t = 1$ **to** $T+1$ **do**
11:   **for all** $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ in descending order **do**
12:     $\alpha(s(\mathbf{z}, \mathcal{P}_t), t) \leftarrow \alpha(s(\mathbf{z}, \mathcal{P}_t), t) - \gamma(\mathbf{z}_{1:|\mathbf{z}|-1}, t-1)$
13:     $\alpha(\mathbf{z}, t) \leftarrow \alpha(\mathbf{z}, t) + \gamma(\mathbf{z}_{\mathbf{1}:|\mathbf{z}|-\mathbf{1}}, t-1)$
14:   **end for**
15:   $\alpha(\epsilon, t) \leftarrow 0$
16:   **for all** $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ in descending order **do**
17:     $\gamma(\mathbf{z}, t) \leftarrow \gamma(\mathbf{z}, t) + \alpha(\mathbf{z}, t)\exp(W(\mathbf{z}, t))$
18:     $\gamma(s(\mathbf{z}, \mathcal{P}_t), t) \leftarrow \gamma(s(\mathbf{z}, \mathcal{P}_t), t) + \gamma(\mathbf{z}, t)$
19:   **end for**
20: **end for**
21: $Z_{\mathbf{x}} \leftarrow \gamma(\epsilon, T+1)$
22: $\delta(\epsilon, T+1) \leftarrow 1$
23: **for** $t = T+1$ **downto** 1 **do**
24:   $\beta(\epsilon, t) \leftarrow \delta(\epsilon, t)$
25:   **for all** $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ in ascending order **do**
26:     $\beta(\mathbf{z}, t) \leftarrow \beta(s(\mathbf{z}, \mathcal{P}_t), t) + \delta(\mathbf{z}, t)$
27:   **end for**
28:   **for all** $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ in descending order **do**
29:     $\delta(\mathbf{z}_{1:t-1}, t-1) \leftarrow \delta(\mathbf{z}_{1:t-1}, t-1) + \beta(\mathbf{z}, t)\exp(W(\mathbf{z}, t)) - \beta(s(\mathbf{z}, \mathcal{P}_t), t)\exp(W(s(\mathbf{z}, \mathcal{P}_t)))$
30:   **end for**
31: **end for**
32: **for** $t = 1$ **to** $T+1$ **do**
33:   **for all** $\mathbf{z} \neq \epsilon \in \mathcal{P}_t$ in descending order **do**
34:     $\theta(\mathbf{z}^p, t) \leftarrow \alpha(\mathbf{z}^p, t)\exp(W(\mathbf{z}^p, t))\beta(\mathbf{z}^p, t)$
35:     $\sigma(\mathbf{z}, t) \leftarrow \sigma(\mathbf{z}, t) + \theta(\mathbf{z}, t)$
36:     $\sigma(s(\mathbf{z}, \mathcal{P}_t), t) \leftarrow \sigma(s(\mathbf{z}, \mathcal{P}_t), t) + \sigma(\mathbf{z}, t)$
37:     **for all** $f_i \in \mathcal{F}_{\mathbf{z},t}$ **do**
38:       $E(f_i) \leftarrow E(f_i) + \sigma(\mathbf{z}, t)/Z_{\mathbf{x}}$
39:     **end for**
40:   **end for**
41: **end for**

を示す．他のものの接尾辞とならない要素 $\mathbf{z}^i$ に対しては，行 13 により $\alpha(\mathbf{z}^i, t) = \gamma(\mathbf{z}_{\mathbf{1}:|\mathbf{z}|-\mathbf{1}}, t)$ となり，式 (16) を満たす．他のものの接尾辞となる要素 $\mathbf{z}^i$ に対しては，それを接尾辞とするものがすべて正しく計算されていれば，行 12 での減算によって，式 (16) が満たされる．「降順」に処理するため，あるラベル列を接尾辞とする他のラベル列は，接尾辞よりも先に計算されている．よって，帰納法によりすべての $\mathbf{z} \in \mathcal{P}_t$ に対して $\alpha(\mathbf{z}, t)$ が正しい値となることが示せる．

その他，類似の証明を省略する．

この手続きの計算量は，1 イテレーション・1 系列ごとに $O\left(\sum_{t=1}^{T+1}\sum_{\mathbf{z}^p:\mathbf{z}^p\in\mathcal{P}_t}|\mathcal{F}_{\mathbf{z}^p,t}| + \sum_{t=1}^{T+1}|\mathcal{P}_t|\right)$ となる．

4.2 デコード

推定されたパラメータを用いて観測列にラベルを付与するデコードのアルゴリズムは，紙面の都合上ここでは述べない．ナイーブな実装では，パラメータ推定の $O(L)$ 倍の計算量となるが，工夫を加えることで $O(\log(L))$ 倍に抑えることができる．

5 実験

CRFSuite[2] を改変したプログラムを使用し，Penn Treebank 3.0 の Wall Street Journal 部分の 0-18 を訓練データ，22-24 をテストデータとして，POS-tagging の実験を行った．結果は表 (1) の通りであり，高次の素性の使用による精度の向上が認められた．それぞれの詳細な条件や，プログラムのソースコード等は `http://vocrf.net/` で公開している．これは Stanford Tagger [3] とも比較可能な精度で

素性関数セット	精度
baseline (1 次 CRF)	96.06%
最大 3 次	96.94%
最大 4 次	97.12%
最大 5 次	97.13%

表 1　WSJ コーパスに対する POS-tagging の精度

あり，本研究の手法によって Linear-Chain CRF の適用範囲が広がる可能性を示唆している．

6 考察・今後の課題

本研究の計算法によって，Linear-Chain CRF において高次の素性を利用することを容易にした．また，POS-tagging というタスクで高次の素性が有効であることを示した．今後，日本語形態素解析への応用を考えたい．

参考文献

[1] J. LAFFERTY. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML, 2001*, 2001.

[2] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[3] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In Proceedings of HLT-NAACL 2003*, pp. 252–259, 2003.

[4] Nan Ye, Wee Sun Lee, Hai Leong Chieu, and Dan Wu. Conditional random fields with high-order features for sequence labeling. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pp. 2196–2204. 2009.