社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

カーネル法に基づく構造データのラベル付け学習アルゴリズム

鹿島　久嗣　　坪井　祐太

† 日本アイ・ビー・エム株式会社 東京基礎研究所　〒 242–8502 神奈川県大和市下鶴間 1623–14
E-mail: †{hkashima,yutat}@jp.ibm.com

あらまし　本稿において，我々はパーセプトロンアルゴリズムに基づく，配列や木，グラフなどの構造を持ったデータのラベル付け学習アルゴリズムを提案する．また，ラベル付けに用いることのできるいくつかのカーネル関数とその効率的な計算法を与える．提案手法は，完全にカーネル化されており，かつ，点ごとのラベル予測を行うため，任意の数の非観測変数を含む，サイズの大きい素性を用いることができる．この点において，提案手法は最大エントロピーマルコフモデルや条件付確率場など少数の非観測変数をもつ素性しか扱うことのできない既存の手法と大きく異なる．
キーワード　カーネル法，パーセプトロン，周辺化カーネル，固有表現抽出，情報抽出

Kernel-based Discriminative Learning Algorithms for Labeling Structured Data

Hisashi KASHIMA and Yuta TSUBOI

† IBM Tokyo Research Laboratory　Shimotsuruma 1623–14, Yamato-shi, Kanagawa, 242–8502 Japan
E-mail: †{hkashima,yutat}@jp.ibm.com

Abstract　We introduce a new perceptron-based discriminative learning algorithm for labeling structural data such as sequences, trees and graphs. Since it is fully kernelized and employs the pointwise label prediction, large features including arbitrary number of hidden variables can be incorporated with polynomial time complexity. This is contrasted with existing labelers that can handle only features of a small number of hidden variables such as Maximum Entropy Markov Models and Conditional Random Fields. We also introduce several kernel functions for labeling sequences, trees and graphs and the efficient algorithms for them.
Key words　Kernel Methods, Perceptron, Marginalized Kernel, Named Entity Recognition, Information Extraction

1. ま え が き

ラベル付け問題は，観測変数集合の値が与えられたときに，非観測変数集合の値を予測する問題であり，教師ありの多クラス分類問題を一般化した問題である．特に，配列ラベル付けは，自然言語処理，バイオインフォマティクス，Web データの解析など多くの分野で見られる重要な問題である．配列ラベル付けの学習問題は長年活発に研究がなされてきたが，より一般的な木やグラフといった構造データのラベル付けについてはほとんど研究がなされていない．本稿では，これら一般的な構造データのラベル付け学習問題において，カーネル法によるアプローチを行う．

配列ラベル付け問題には長年，隠れマルコフモデル (HMM) が用いられ，一定の成功を収めてきた．しかしながら，HMM は観測変数と非観測変数の同時確率分布を推定するという，より難しい問題を解いており，多くの学習データを必要とする．また，観測変数は，非観測変数が与えられると互いに独立であることが求められるため，単語と，その頭文字のように互いに独立でない素性を自然に扱うことができないという問題がある．一方，近年，最大エントロピーマルコフモデル (MEMM) [1] や条件付確率場 (CRF) [2] などの条件付モデルと呼ばれるモデルが，観測変数から非観測変数を予測するという目的に適しているため，注目を集めている．さらに，Collins [3] は隠れマルコフ (HM) パーセプトロンと呼ばれる，パーセプトロンアルゴリズムに基づく効率的なアルゴリズムを提案している．これを拡張する形で，サポートベクターマシン (SVM) に基づくアルゴリズム [4] やブースティングに基づくアルゴリズム [5] も提案されている．

いくつかの問題において，自然言語処理におけるイディオムや，バイオインフォマティクスにおけるモチーフなどのようにバイグラムなどの局所的な素性を組み合わせるだけでは広い範囲の文脈を捉えるのに充分でない場合がある．しかしながら，現在提案されているほとんどの手法では，少数の非観測変数を含む素性しか用いることができない．ラベル付けの段階は，通

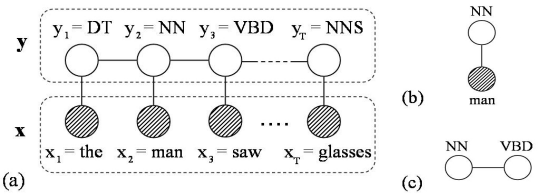

図 1 (a) 1 次のマルコフ仮定に基づく配列のグラフィカルモデル表現. 黒いノードは観測変数を，白いノードは非観測変数を表す．文 “the, man, saw ,· · · , glasses.”は，観測変数のラベル列 $\boldsymbol{x}$，品詞タグ列 “DT, NN, VBD, . . ., NNS”は，非観測変数のラベル列 $\boldsymbol{y}$ を表す．(b) 観測変数と非観測変数のペアからなる素性 (c) 連続する 2 つの非観測変数からなる素性

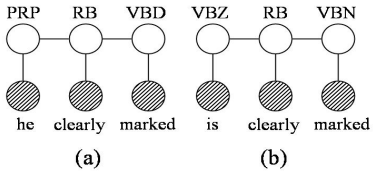

図 2 配列ラベリングのための素性 (長さ 3)．バイグラムの素性では”marked”の品詞を決定するのには充分でない．

常 Viterbi アルゴリズムのような動的計画法に基づく方法が用いられるが，ひとつの素性に含まれる非観測変数の数に対して計算量が指数的に増大する．HM パーセプトロンや HMSVM は双対型で書くこともできるが，やはりこの問題は解決されない．

本稿では，はじめにラベル付け候補を絞ることなく，大局的な素性を用いてラベル付けを行うことのできる方法である**周辺化ラベル付けパーセプトロン**を提案する．これは，周辺化カーネル [6] と，点ごとのラベル予測 [7] という 2 つのアイデアに基づいている．点ごとのラベル予測によって，各非観測変数のラベルをそれぞれ予測できるため，前述の問題が解決される．また，周辺化カーネル [6] を用いたカーネル化によって，任意の数の非観測変数を含むような大きい素性を扱うことができるようになる．さらに我々は，配列や順序木といったいくつかの構造データにたいして，実際の周辺化カーネル関数を提案する．近年，カーネル法を用いた構造データの分類は精力的に研究されており，配列カーネル [8], [9] や木カーネル [10], [11]，グラフカーネル [12], [13] などが提案されている(注1)．これらは大抵，広範囲にわたる文脈を取り入れるため，部分列や部分グラフなどの部分構造が，構造データ内に出現する回数などを用いて素性ベクトルを定義している．これらの素性ベクトルは通常，高次元 (場合によっては無限次元) になるため，接尾辞木や動的計画法，あるいは行列計算などを用いて，素性を陽に数え上げることなく効率的にカーネル関数を計算する方法を提案されている．我々はこれらのカーネルに基づき，配列や順序木に対する周辺化カーネルと，その効率的な計算方法を与える．(注2)
また，最後に実世界データを用いた，配列と木に対するラベル付け問題の実験を行い，その有効性を検証する．

2. 隠れマルコフ (HM) パーセプトロン

ラベル付け問題は，観測変数 $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ から非観測変数 $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ へのマッピングを求める問題として定義できる．ここで $x_t \in \Sigma_x$，$y_t \in \Sigma_y$ とする．例えば，品詞タグ付けなどのタスクにおいて，x_t は t 番目の単語を表わし，y_t は t 番目の単語の品詞を表す (図 1)．学習器は，訓練データの集合 $E = (e^{(1)}, e^{(2)}, \ldots, e^{(|E|)})$ を用いることができる．ここで，$e^{(i)} = (\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ は i 番目の訓練例で，$|\boldsymbol{x}^{(i)}| = |\boldsymbol{y}^{(i)}| = T^{(i)}$ とする．

Collins [3] は，CRF や MEMM の代替手法としてパーセプトロンに基づくラベル付け学習アルゴリズムである HM パーセプトロンを提案した．HM パーセプトロンは訓練例をオンライン型で順番に処理できるため効率がよく，その性能は CRF に匹敵することが示されている [16]．HM パーセプトロンの重要な点は，観測変数集合から非観測変数集合へのマッピングを，観測変数集合の値と非観測変数集合の値の組の 2 値分類，すなわち ${\Sigma_x}^T \times {\Sigma_y}^T \to \{+1, -1\}$ として捉える点である．F を全ての素性の集合とし，$\phi_f(\boldsymbol{x}, \boldsymbol{y})$ を，素性 $f \in F$ が $(\boldsymbol{x}, \boldsymbol{y})$ 内に出現する回数，$\Phi(\boldsymbol{x}, \boldsymbol{y})$ はそれをベクトルの形にあらわしたものとする．$\boldsymbol{x}$ が入力として与えられると，HM パーセプトロンは現在の素性の重みに基づき，予測 $\hat{\boldsymbol{y}}$ を出力する．

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in {\Sigma_y}^T} \sum_{f \in F} w_f \phi_f(\boldsymbol{x}, \boldsymbol{y}) = \operatorname*{argmax}_{\boldsymbol{y} \in {\Sigma_y}^T} \left\langle \boldsymbol{w}, \Phi(\boldsymbol{x}, \boldsymbol{y}) \right\rangle \quad (1)$$

ここで w_f は素性 f の重みであり，$\boldsymbol{w}$ はこれらをベクトルの形で表したものとする．1 次のマルコフ仮定の下では，観測変数と非観測変数のペアからなる素性 (図 1(b)) と，連続する 2 つの非観測変数からなる素性 (図 1(c)) の，2 つのタイプの素性が考えられる．重みの学習は，訓練例 $e^{(i)}$ に対する予測が誤り，すなわち $\hat{\boldsymbol{y}}^{(i)} \neq \boldsymbol{y}^{(i)}$ であったときに，

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} + \Phi(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \Phi(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) \quad (2)$$

によって更新される．さらに (1) は訓練例の重み α を用いることで双対型で書くことができる．

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in {\Sigma_y}^T} \sum_{j=1}^{|E|} \sum_{\tilde{\boldsymbol{y}} \in {\Sigma_y}^T} \alpha_j(\tilde{\boldsymbol{y}}) \left\langle \Phi(\boldsymbol{x}^{(j)}, \tilde{\boldsymbol{y}}), \Phi(\boldsymbol{x}, \boldsymbol{y}) \right\rangle \quad (3)$$

初期値 $\boldsymbol{\alpha} = \boldsymbol{0}$ から始めると，重みの更新ルールは以下のように書き直せる．

- $\alpha_i^{new}(\boldsymbol{y}^{(i)}) = \alpha_i^{old}(\boldsymbol{y}^{(i)}) + 1$
- $\alpha_i^{new}(\hat{\boldsymbol{y}}^{(i)}) = \alpha_i^{old}(\hat{\boldsymbol{y}}^{(i)}) - 1$

$(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{y}}^{(i)})$ は負例として働くため，擬似負例と言われる．

さて，問題によっては上で用いたサイズ 2 の素性 (図 1(b)(c)) では広範囲にわたる文脈をモデル化するのには充分でない場合がある．そこで HM パーセプトロンにおいて，長さ d までの素性 (図 2) を用いることを考える．$\boldsymbol{x}_{t_1}^{t_2}$ は $\boldsymbol{x}$ の t_1 番目から t_2 番目までの観測変数集合を表すとし，$\boldsymbol{y}_{t_1}^{t_2}$ も同様に定義する．全ての素性は高々 d 個の非観測変数に依存することを考慮すると，$\Phi(\boldsymbol{x}, \boldsymbol{y})$ は以下のように展開できる．

$$\begin{aligned} \Phi(\boldsymbol{x}, \boldsymbol{y}) = \; & \Phi(\boldsymbol{x}_1^d, \boldsymbol{y}_1^d) - \Phi(\boldsymbol{x}_2^d, \boldsymbol{y}_2^d) \\ & + \Phi(\boldsymbol{x}_2^{d+1}, \boldsymbol{y}_2^{d+1}) - \Phi(\boldsymbol{x}_3^{d+1}, \boldsymbol{y}_3^{d+1}) \\ & \cdots \\ & + \Phi(\boldsymbol{x}_{T-d}^{T-1}, \boldsymbol{y}_{T-d}^{T-1}) - \Phi(\boldsymbol{x}_{T-d+1}^{T-1}, \boldsymbol{y}_{T-d+1}^{T-1}) \\ & + \Phi(\boldsymbol{x}_{T-d+1}^{T}, \boldsymbol{y}_{T-d+1}^{T}). \end{aligned} \quad (4)$$

(注1)：[14] に詳細なサーベイがある．
(注2)：非循環有向グラフに対する周辺化カーネルは [15] を参照されたい．

(4) を (1) に代入することで，$\boldsymbol{y}$ の予測は動的計画法を用いて行うことができる．

$$\begin{aligned}\max_{\boldsymbol{y}}\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y})\big\rangle &= \max_{y_1,\ldots,y_d} s(y_1,\ldots,y_d)\\ s(y_t,\ldots,y_{t+d-1}) &= \max_{y_{t+d}} s(y_{t+1},\ldots,y_{t+d})\\ &\quad +\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_t^{t+d-1},\boldsymbol{y}_t^{t+d-1}) - \Phi(\boldsymbol{x}_{t+1}^{t+d-1},\boldsymbol{y}_{t+1}^{t+d-1})\big\rangle\\ s(y_{T-d+1},\ldots,y_T) &= \big\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_{T-d+1}^{T},\boldsymbol{y}_{T-d+1}^{T})\big\rangle\end{aligned}$$

しかしながら，各再帰のステップの計算時間は d について指数的に増加してしまう．たとえ双対型 (3) にしても，長さ d までの非観測変数の文脈は陽に考えなければならないため同様の問題は残る．これはカーネルを用いて任意のサイズの素性を導入する上で重大な障害となる．CRF も同様の分解に基づく動的計画法を用いるため同様である．さらには，素性の出現を数える際ギャップなどの曖昧さを許すようなカーネル [8] を用いるに到っては，長さ d の素性が，長さ d 以上の領域にまたがりうるため，分解 (4) でさえ用いることができない．Collins は，あらかじめ学習された，局所的な素性のみを用いる別のラベル付け器を用いて，適当な数のラベル付け候補を出力し，それをより広範囲にわたる素性を用いて再ランキングするという方法でこの問題を避けている [17] が，正しいラベル付けが，広範囲の素性に基づく始めの候補の中に含まれているという保証はない．

3. 周辺化ラベル付けパーセプトロン

前節で述べた問題点を解決するために，カーネル法に基づくラベル付け器を提案する．通常，MEMM や CRF は，モデル $P_s(\boldsymbol{y}|\boldsymbol{x})$ は対数尤度の和 $\sum_i \log P_s(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)})$ を最大化することによって学習する．この目的関数は，ひとつの配列上の全部の非観測変数を正しくラベル付することを目指しているといえる．一方，Kakade ら [7] は多くのタスクにおいて，個々の非観測関数のラベルを正しくラベル付けできればよいという点に注目し，点ごとに周辺化された尤度に基づく目的関数 $\sum_i \sum_t \log P_p(y_t = y_t^{(i)}|\boldsymbol{x}^{(i)})$ を提案した．ここで，

$$P_p(y_t = y_t^{(i)}|\boldsymbol{x}^{(i)}) = \sum_{\boldsymbol{y}:y_t=y_t^{(i)}} P_s(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)})$$

は，t 番目の非観測変数のラベルを $\hat{y}_t \in \Sigma_y$ に固定し，それ以外の非観測変数について周辺化した $P_s(\boldsymbol{y}|\boldsymbol{x})$ である．この目的関数は，もともとの配列ごとの尤度に基づく目的関数に匹敵する性能をもつことが示されている [16]．ここで重要な点は，P_p は y_t にのみ依存するため，ラベル予測も点ごとに行われるということである．我々はこのアイデアと，パーセプトロンに基づくラベル付け器とを組み合わせることで，以下に定義される**周辺化ラベル付けパーセプトロン**を提案する．

$$\hat{y}_t = \operatorname*{argmax}_{\tilde{y}_t\in\Sigma_y} \sum_{\boldsymbol{y}:y_t=\tilde{y}_t} P(\boldsymbol{y}|\boldsymbol{x})\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y})\big\rangle \tag{5}$$

argmax 操作は $\tilde{y}_t$ にのみ依存するので，点ごとのラベル予測が実現されている．基本的な考え方としては以下のとおりである．y_t を予測する際，もともとのラベル付けパーセプトロンの出力 $\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y})\rangle$ は，t 番目の非観測変数のラベルを $\hat{y}_t$ に固定し，そのほかの非観測変数について周辺化される．$P(\boldsymbol{y}|\boldsymbol{x})$ は，非観測変数についての事前分布であり，これはあらかじめ訓練された MEMM や CRF などでもよい．また，周辺化された素性ベクトル $\sum_{\boldsymbol{y}:y_t=\tilde{y}_t} P(\boldsymbol{y}|\boldsymbol{x})\Phi(\boldsymbol{x},\boldsymbol{y})$ は，新しい素性ベクトルと考えることができるので，$e^{(i)}$ の t 番目の非観測変数についての予測が誤りだったとき，すなわち，$\hat{y}_t^{(i)} \neq y_t^{(i)}$ のとき，重みベクトル $\boldsymbol{w}$ は，以下の更新則によって更新される．

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} + \sum_{\boldsymbol{y}:y_t=y_t^{(i)}} P(\boldsymbol{y}|\boldsymbol{x})\Phi(\boldsymbol{x},\boldsymbol{y}) - \sum_{\boldsymbol{y}:y_t=\hat{y}_t^{(i)}} P(\boldsymbol{y}|\boldsymbol{x})\Phi(\boldsymbol{x},\boldsymbol{y}) \tag{6}$$

次に，周辺化ラベル付けパーセプトロンの双対型を導くことで，これをカーネル化する．$\phi_f(\boldsymbol{x},\boldsymbol{y};t)$ を，素性 f が $(\boldsymbol{x},\boldsymbol{y})$ 内に，$(\boldsymbol{x},\boldsymbol{y})$ の t 番目の非観測変数を含む形で出現する回数とし，また $\Phi_f(\boldsymbol{x},\boldsymbol{y};t)$ をそのベクトル表現とする．まず，もともとの形 (5) を $\tilde{y}_t$ に依存しない項を足すことで以下のように変形する．

$$\begin{aligned}\hat{y}_t &= \operatorname*{argmax}_{\tilde{y}_t\in\Sigma_y} \sum_{\boldsymbol{y}:y_t=\tilde{y}_t} P(\boldsymbol{y}|\boldsymbol{x})\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y})\big\rangle\\ &\quad + \sum_{\tilde{y}_t\in\Sigma_y}\sum_{\boldsymbol{y}:y_t=\tilde{y}_t} P(\boldsymbol{y}|\boldsymbol{x})\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y};t)\big\rangle\\ &\quad - \sum_{\boldsymbol{y}\in{\Sigma_y}^T} P(\boldsymbol{y}|\boldsymbol{x})\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y})\big\rangle\\ &= \operatorname*{argmax}_{\tilde{y}_t\in\Sigma_y} \sum_{\boldsymbol{y}:y_t=\tilde{y}_t} P(\boldsymbol{y}|\boldsymbol{x})\big\langle \boldsymbol{w}, \Phi(\boldsymbol{x},\boldsymbol{y};t)\big\rangle\end{aligned} \tag{7}$$

更新則は以下のように書き換えられる．

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} + \sum_{\boldsymbol{y}:y_t=y_t^{(i)}} P(\boldsymbol{y}|\boldsymbol{x})\Phi(\boldsymbol{x},\boldsymbol{y};t) - \sum_{\boldsymbol{y}:y_t=\hat{y}_t^{(i)}} P(\boldsymbol{y}|\boldsymbol{x})\Phi(\boldsymbol{x},\boldsymbol{y};t)$$

例の重み $\boldsymbol{\alpha}$ を導入して，$\boldsymbol{w}$ を線形和の形で書くと，

$$\boldsymbol{w} = \sum_{j=1}^{|E|}\sum_{\tau=1}^{T^{(j)}}\sum_{\breve{y}_\tau\in\Sigma_y} \alpha_{j\tau}(\breve{y}_\tau) \sum_{\boldsymbol{y}:y_\tau=\breve{y}_\tau} P(\boldsymbol{y}|\boldsymbol{x})\Phi(\boldsymbol{x}^{(j)},\boldsymbol{y};\tau) \tag{8}$$

となるので，これを (7) に代入することで，**周辺化カーネルラベル付けパーセプトロン**

$$\hat{y}_t = \operatorname*{argmax}_{\tilde{y}_t\in\Sigma_y} \sum_{j=1}^{|E|}\sum_{\tau=1}^{T^{(j)}}\sum_{\breve{y}_\tau\in\Sigma_y} \alpha_{j\tau}(\breve{y}_\tau) K(\boldsymbol{x}^{(j)},\boldsymbol{x},\tau,t,\breve{y}_\tau,\tilde{y}_t) \tag{9}$$

を得る．ここで，

$$\begin{aligned}&K(\boldsymbol{x}^{(j)},\boldsymbol{x},\tau,t,\breve{y}_\tau,\tilde{y}_t) = \\ &\sum_{\boldsymbol{y}:y_\tau=\breve{y}_\tau}\sum_{\boldsymbol{y}':y'_t=\tilde{y}_t} P(\boldsymbol{y}|\boldsymbol{x}^{(j)})P(\boldsymbol{y}'|\boldsymbol{x})\big\langle \Phi(\boldsymbol{x}^{(j)},\boldsymbol{y};\tau), \Phi(\boldsymbol{x},\boldsymbol{y}';t)\big\rangle\end{aligned} \tag{10}$$

は 2 つの例の τ 番目と t 番目の非観測変数のラベルをそれぞれ固定し周辺化されたカーネル関数であるため周辺化カーネル [6] と呼ばれる．アルゴリズムを図 3 に示す．

4. 周辺化カーネル関数の計算

本節では配列及び順序木のラベル付けのための周辺化カーネル (10) を提案する．近年，さまざまな構造データに対するカーネル関数は精力的に研究されている [14] が，これをラベル付けに使えるよう拡張する．ここでは簡単のため，事前分布 $P(\boldsymbol{y}|\boldsymbol{x})$ は，$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{T} P(y_t|x_t)$ のように分解できるとするが，本稿の結果は CRF などのより一般的なモデルを事前分布として用いる場合にも拡張可能である．また，表記上の簡単

```
// Initialization
α := 0
// Training
for round = 1, ..., max_round,
  for i = 1, ..., |E|,
    // kernel computation
    for j = 1, ..., |E|,
      for t = 1, ..., T^(i), and τ = 1, ..., T^(j),
        for y̆_τ ∈ Σ_y, and ỹ_τ ∈ Σ_y,
          compute K(x^(j), x^(i), τ, t, y̆_τ, ỹ_t)
    // prediction & update
    for t = 1, ..., T^(i),
      predict ŷ_t^(i) by using Equation (9)
      if ŷ_t^(i) ≠ y_t^(i),
        α_it(y_t^(i)) := α_it(y_t^(i)) + 1
        α_it(ŷ_t^(i)) := α_it(ŷ_t^(i)) − 1
```

図 3　周辺化カーネルラベル付けパーセプトロン

のため本節の残りの部分では，$K(\tau, t, \breve{y}_\tau, \tilde{y}_t)$ を i 番目の例と j 番目の例が与えられたときのカーネル関数とする．我々のカーネル関数を計算する上で重要な点は，すべての素性が，より小さい素性を組み合わせることで表現できる点である．たとえば，図 4 で素性 (b) が，その一番右端を t 番目の非観測変数に，また素性 (c) が，その一番左端を t 番目の非観測変数に置いて出現しているとすると，素性 (b) と (c) を組み合わせた素性 (a) は，その 2 番目の変数を t 番目の非観測変数に置いて出現していることになる．従って，本節で提案する周辺化カーネルはすべて上流カーネル $K_U(\tau, t)$，下流カーネル $K_D(\tau, t)$，点カーネル $K_P(\tau, t, \breve{y}_\tau, \tilde{y}_t)$ の 3 つに分解できるとする．

$$K(\tau, t, \breve{y}_\tau, \tilde{y}_t) = K_U(\tau, t) \cdot K_P(\tau, t, \breve{y}_\tau, \tilde{y}_t) \cdot K_D(\tau, t)$$

$$K_U(\tau, t) = \sum_{\boldsymbol{y}_U(\tau)} \sum_{\boldsymbol{y}'_U(t)} P(\boldsymbol{y}_U(\tau)|\boldsymbol{x}_U^{(j)}(\tau)) P(\boldsymbol{y}'_U(t)|\boldsymbol{x}_U^{(i)}(t)) \cdot \Big\langle \Phi(\boldsymbol{x}_U^{(j)}(\tau), \boldsymbol{y}_U(\tau); \tau), \Phi(\boldsymbol{x}_U^{(i)}(t), \boldsymbol{y}'_U(t); t) \Big\rangle \quad (11)$$

$$K_D(\tau, t) = \sum_{\boldsymbol{y}_D(\tau)} \sum_{\boldsymbol{y}'_D(t)} P(\boldsymbol{y}_D(\tau)|\boldsymbol{x}_D^{(j)}(\tau)) P(\boldsymbol{y}'_D(t)|\boldsymbol{x}_D^{(i)}(t)) \cdot \Big\langle \Phi(\boldsymbol{x}_D^{(j)}(\tau), \boldsymbol{y}_D(\tau); \tau), \Phi(\boldsymbol{x}_D^{(i)}(t), \boldsymbol{y}'_D(t); t) \Big\rangle \quad (12)$$

$$K_P(\tau, t, \breve{y}_\tau, \tilde{y}_t) = \frac{P(\breve{y}_\tau|x_\tau^{(j)}) P(\tilde{y}_t|x_t^{(i)}) \big\langle \Phi(x_\tau^{(j)}, \breve{y}_\tau; \tau), \Phi(x_t^{(i)}, \tilde{y}_t; t) \big\rangle}{\Big(\sum_{\breve{y}_\tau} \sum_{\tilde{y}_t} P(\breve{y}_\tau|x_\tau^{(j)}) P(\tilde{y}_t|x_t^{(i)}) \big\langle \Phi(x_\tau^{(j)}, \breve{y}_\tau; \tau), \Phi(x_t^{(i)}, \tilde{y}_t; t) \big\rangle \Big)^2} \quad (13)$$

ここで，$\boldsymbol{y}_U(\tau)$ と $\boldsymbol{y}_D(\tau)$ は，それぞれ τ 番目から上流あるいは下流に位置する非観測変数の集合とする ($\boldsymbol{y}_U(\tau) \cap \boldsymbol{y}_D(\tau) = \{y_\tau\}$)．また $K_P(\tau, t, \breve{y}_\tau, \tilde{y}_t)$ の分母は $K_U(\tau, t)$ と $K_D(\tau, t)$ の重なり部分を打ち消すための項である．K_U や K_D における周辺化を陽に計算することは現実的ではないが，多くの場合，動的計画法などをつかうことで効率的に計算可能である．この分解によって，i 番目と j 番目の例が与えられたときに，全ての $(\tau, t, \breve{y}_\tau, \tilde{y}_t)$ のとりうる値に対し周辺化カーネルを 1 回の計算で計算できる．

4.1　配列のラベル付けのための周辺化カーネル

ここでは配列における広範囲にわたる依存関係を考慮するため，素性は図 2 に示すような，任意の長さ連結した非観測変数と観測変数のペアとする．素性のサイズに制限を設けないため，様々なサイズの素性が入り混じることになるが，そこで，混合定数 $c > 0$ を用いて，ϕ_f を属性 f のサイズ d に応じて重み付ける．例えば f が $(\boldsymbol{x}, \boldsymbol{y})$ に N 回出現したとすると，$\phi_f(\boldsymbol{x}, \boldsymbol{y})$ は $N \cdot c^d$ と定義する．$K_U(\tau, t)$ や $K_D(\tau, t)$ において，ある点 τ より上流の変数 $\boldsymbol{y}_U(\tau)$ とは，その点とそれより左にある変数のことである．同様に下流の変数 $\boldsymbol{y}_D(\tau)$ とは，その点とそれより右にある変数のことであるとする．アルゴリズムは，丁度 HMM の前向き・後ろ向きアルゴリズムのように，それぞれ K_U と K_D を計算するための 2 つの動的計画法のループからなる．

$$K_U(\tau, t) = c^2 k(x_\tau^{(j)}, x_t^{(i)}) (K_U(\tau - 1, t - 1) + 1) \quad (14)$$

$$K_D(\tau, t) = c^2 k(x_\tau^{(j)}, x_t^{(i)}) (K_D(\tau + 1, t + 1) + 1) \quad (15)$$

ここで，

$$k(x_\tau^{(j)}, x_t^{(i)}) = \sum_{\breve{y}_\tau \in \Sigma_y} \sum_{\tilde{y}_t \in \Sigma_y} P(\breve{y}_\tau|x_\tau^{(j)}) P(\tilde{y}_t|x_t^{(i)}) \cdot k_y(\breve{y}_\tau, \tilde{y}_t) k_x(x_\tau^{(j)}, x_t^{(i)})$$

とする．境界条件は $K_U(\tau, 0) = K_U(0, t) = K_D(\tau, |T^{(i)}| + 1) = K_D(|T^{(j)}| + 1, t) = 0$ である．K_P は以下によって計算できる．

$$K_P(\tau, t, \breve{y}_\tau, \tilde{y}_t) = \frac{c^2 P(\breve{y}_\tau|x_\tau^{(j)}) P(\tilde{y}_t|x_t^{(i)}) k_y(\breve{y}_\tau, \tilde{y}_t) k_x(x_\tau^{(j)}, x_t^{(i)})}{\big(c^2 k(x_\tau^{(j)}, x_t^{(i)})\big)^2} \quad (16)$$

ここで，k_x と k_y は 2 つの引数が同一であるときに 1 を，そうでないときに 0 を返す関数とするが，ここを 2 つの変数の間のカーネル関数で置き換えることも可能である．明らかに，このカーネル関数の計算量は $O(T^{(i)} T^{(j)})$ である．(注3)

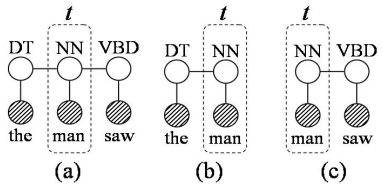

図 4　上流の素性と下流の素性を組み合わせることで，大きい素性が構成される例

4.2　順序木のラベル付けのための周辺化カーネル

順序木 (図 5) のラベル付けのためのカーネルを，順序木カーネル [11] を拡張することによって設計する．素性は図 5 に示すような任意の順序木の形をしているとする．配列ラベル付けのときと同様，混合定数 c を用いて $\phi_f(\boldsymbol{x}, \boldsymbol{y})$ を定義する．$K_U(\tau, t)$ や $K_D(\tau, t)$ において，ある点 τ より上流の変数 $\boldsymbol{y}_U(\tau)$ とは，その点とそれより外側にある変数のことであり，下流の変数 $\boldsymbol{y}_D(\tau)$ とは，その点とそれより内側にある変数のことであるとする．アルゴリズムは，丁度，確率文脈自由文法の内側・外側アルゴリズムのように，それぞれ K_U と K_D を計算するための 2 つの動的計画法のループからなる．$ch(\tau, \upsilon)$ を τ 番目のノー

(注3)：ここでの結果は [8] にみられるようなギャップを許した曖昧なマッチングを許すように拡張可能である．詳しくは [15] を参照されたい．

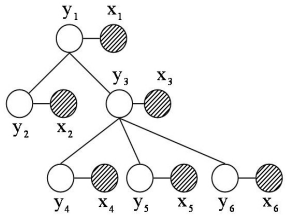

図 5　順序木の例：黒いノードは観測変数を，白いノードは非観測変数を表す.

ドの v 番目の子を表すインデクスとし，$pa(\tau)$ を τ 番目のノードの親ノードを表すインデクスとする．また，$\#ch(\tau)$ を τ 番目のノードの子供の数とし，$chID(\tau)$ を，τ 番目のノードがその親ノードの何番目の子ノードであるかを表すインデクスとする．

順序木カーネル [11] の再帰的計算と同様にして，K_D は後行順で動的計画法により，

$$K_D(\tau,t) = c^2 k(x_\tau^{(j)}, x_t^{(i)}) S_F(\tau,t,\#ch(\tau),\#ch(t))$$

によって計算される．ここで S_F も以下によって再帰的に定義される．

$$\begin{aligned} S_F(\tau,t,v,u) = \; & K_D(ch(\tau,v),ch(t,u)) S_F(\tau,t,v-1,u-1) \\ & + S_F(\tau,t,v-1,u) + S_F(\tau,t,v,u-1) \\ & - S_F(\tau,t,v-1,u-1) \end{aligned}$$

境界条件は $S_F(\tau,t,0,u) = S_F(\tau,t,v,0) = 1$ である．直感的には $S_F(\tau,t,v,u)$ は $ch(\tau,1)$ 番目から $ch(\tau,v)$ 番目の子集合と，$ch(t,1)$ 番目から $ch(t,u)$ 番目まで子集合の全ての，順序を保存したマッチングについての和をとっているといえる．

同時に，あとで K_U を計算するときのために，$ch(\tau,v)$ 番目から $ch(\tau,\#ch(\tau))$ 番目の子集合と，$ch(t,u)$ 番目から $ch(t,\#ch(\tau))$ 番目まで子集合の全てのマッチングについての和をとったものを表す $S_B(\tau,t,v,u)$ を

$$\begin{aligned} S_B(\tau,t,v,u) = \; & K_D(ch(\tau,v),ch(t,u)) S_B(\tau,t,v+1,u+1) \\ & + S_B(\tau,t,v+1,u) + S_B(\tau,t,v,u+1) \\ & - S_B(\tau,t,v+1,u+1) \end{aligned}$$

と定義する．境界条件は $S_B(\tau,t,\#ch(\tau)+1,u) = S_B(\tau,t,v,\#ch(t)+1) = 1$ である．

K_U は先行順の動的計画法で計算される．$K_U(\tau,t)$ は，親ノード同士の上流カーネルと，兄弟同士の下流カーネルを組み合わせることで計算できる (図 6).

$$\begin{aligned} K_U(\tau,t) = \; & c^2 k(x_\tau^{(j)}, x_t^{(i)}) \Big(1 + K_U(pa(\tau),pa(t)) \\ & \cdot S_F(pa(\tau),pa(t),chID(\tau)-1,chID(t)-1) \\ & \cdot S_B(pa(\tau),pa(t),chID(\tau)+1,chID(t)+1) \Big) \end{aligned}$$

K_P は (16) と同様である．[11] と同様の解析によってこのカーネルの計算量は $O(T^{(i)}T^{(j)})$ であることが容易に分かる．

5. 実　　験

本節では自然言語処理の課題 (固有表現抽出と製品使用情報抽出) での実験結果を示し，提案手法により効率的に扱うことが可能になった大きな構造素性の有用性を検証する．本実験では周辺化ラベル付けパーセプトロンと HM パーセプトロンの性能比較を行った.

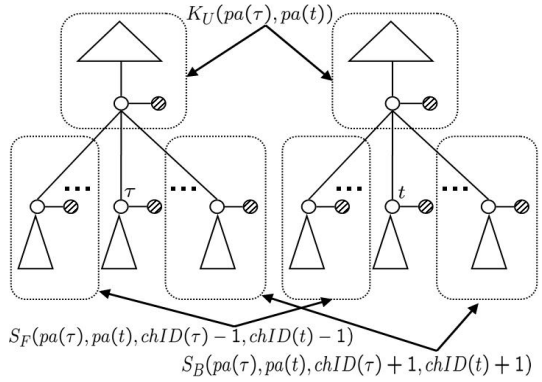

図 6　順序木における $K_U(\tau,t)$ の計算のイメージ.

表 1　固有表現抽出結果 (括弧内は標準偏差)

	精度	適合率	再現率	F1
配列カーネル	**88.7**% (3.4)	**49.0**% (6.0)	**23.1**% (8.1)	**30.5** (6.7)
HM パーセプトロン	80.2% (11.5)	23.8% (14.6)	17.9% (3.0)	18.6 (5.2)

表 2　製品使用情報抽出結果 (括弧内は標準偏差)

	精度	適合率	再現率	F1
Sequence Kernel	89.7% (2.0)	**52.2**% (9.5)	29.6% (4.0)	37.5(4.1)
Tree Kernel	**89.9**% (2.7)	51.4% (10.9)	**32.5**% (14.4)	**38.9** (12.1)
HM-Perceptron	89.7%(1.8)	51.5%(8.5)	24.0%(21.4)	28.9(20.3)

5.1　固有表現抽出

固有表現抽出は情報抽出課題の一つで，文書内の人名・地名・組織名などを識別する問題である．実験では，CoNLL2002 固有表現抽出課題で提供されたスペイン語のデータの最初の 300 文 (8,541 単語) を使用した．この課題は，各単語に固有表現の種類を示すラベルを付与する課題である．ラベルの数は，各固有表現の種類と表現の境界種別の組み合わせ 8 ラベルと非固有表現を示すダミーラベルを合わせた 9 種類 ($|\Sigma_y| = 9$) である．観測変数のカーネルは単語とそのスペリング (詳細は [16] の S2 素性を参照のこと) を素性とした 2 次の多項式カーネルを使用した．ただし，HM パーセプトロンは [4] に倣い前後 1 単語の素性も含めた．周辺化ラベル付けパーセプトロンでは配列ラベル付けカーネル (配列カーネル) を用いた．混合定数は事前実験により求めた $c = 1$ とした．また，非観測変数に対する事前分布は一様分布 ($P(y_t|x_t) = 1/|\Sigma_y|$) として設計した．性能評価にはダミーラベルを含めたラベル付けの精度，固有表現ラベルの適合率・再現率，適合率と再現率の調和平均値である F1 尺度を用いた.

表 1 は 5 分割交差検定の結果である．提案手法は HM パーセプトロンを大きく上回る性能を示した．配列ラベル付けカーネルを用いた提案手法が良い性能を示した理由は，固有表現抽出の性質によるものと考えられる．固有表現抽出課題は自然言語処理の問題の中で広い文脈情報を考慮する必要がある問題の一つである．例えば，「ホワイトハウス」は文脈により場所または組織を示すかが異なる．この実験結果は，人手による素性の選択なしに幅広い文脈情報を効率的に扱うことが出来るカーネル化されたラベル付け学習アルゴリズムの優位性を示すものと言える.

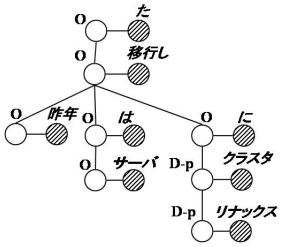

図 7　文 “昨年サーバはリナックスクラスタに移行した” の依存構造木．非観測変数 “D-p” は顧客の使用している製品名，“O” はダミーラベルを示す．

5.2　製品使用情報抽出

製品使用情報抽出は営業記録から顧客の製品使用状況を抽出する固有表現抽出の特殊な課題である．実験では日本語の営業記録 184 文 (3,570 語) を用いた．この課題では単語に対して，顧客が導入している製品・製造企業名・導入数・購入理由と導入済み・検討中・導入却下という導入状況の組み合わせのラベルの付与を目的とした．ダミーラベルを含め使用データでのラベル数は 12 である ($|\Sigma_y| = 12$)．日本語文は形態素解析器により単語に分割し品詞と正規形を付与した．観測変数のカーネルは単語・品詞・正規形・文字種を素性とした 2 次の多項式カーネルを使用した．ただし，HM パーセプトロンは前後 1 単語の素性も含めた．周辺化ラベル付けパーセプトロンでは配列カーネルと順序木のラベル付けカーネル (木カーネル) を用いた．木カーネルでは単語間の依存関係を示す依存構造木に対するラベル付け問題とした．図 7 に依存構造木の例を示す．文の依存構造木の生成には統計パーザ [18] を用いた．(注4) 両カーネルとも混合定数は $c = 0.8$ とした．

5 分割交差検定の結果を図 2 に示す．固有表現抽出課題と同様に HM パーセプトロンに比べて提案手法は優れた性能を示した．また，総合的には木カーネルが若干良い性能を示した．この結果は製品使用情報抽出において製品名と字面としては離れている用言との依存関係が重要であるという直感に合致しており，構造的な情報の有用性を示していると言える．

6. おわりに

パーセプトロンアルゴリズムに基づく，配列や木，グラフなどの構造を持ったデータのラベル付け学習アルゴリズムを提案した．また，配列や順序木のラベル付けに用いることのできる周辺化カーネル関数とその効率的な計算法を提案し，実際のデータを用いた実験により，その有効性を検証した．今後の発展としては，CRF や MEMM などのより複雑な事前分布の導入，HM-SVM [4] のような SVM 化などが考えられる．

文　献

[1] A. McCallum, D.Freitag, and F. Pereira, “Maximum entropy Markov models for information extraction and segmentation,” Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, pp.591–598, Morgan Kaufmann, 2000.

[2] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, pp.282–289, Morgan Kaufmann, 2001.

[3] M. Collins, “Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms,” Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2002.

[4] Y. Altun, I. Tsochantaridis, and T. Hofmann, “Hidden Markov support vector machines,” Proceedings of the Twentieth International Conference on Machine Learning, 2003.

[5] Y. Altun, T. Hofmann, and M. Johnson, “Discriminative learning for label sequences via boosting,” Advances in Neural Information Processing Systems 15, ed. S. Becker, S. Thrun, and K. Obermayer, Cambridge, MA, MIT Press, 2003.

[6] K. Tsuda, T. Kin, and K. Asai, “Marginalized kernels for biological sequences,” Bioinformatics, vol.18, no.Suppl. 1, pp.S268–S275, 2002.

[7] S. Kakade, Y.W. Teh, and S. Roweis, “An alternative objective function for Markovian fields,” Proceedings of the Nineteenth International Conference on Machine Learning, San Francisco, CA, pp.275–282, Morgan Kaufmann, 2002.

[8] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text classification using string kernels,” Journal of Machine Learning Research, vol.2, pp.419–444, 2002.

[9] C. Leslie, E. Eskin, and W.S. Noble, “The spectrum kernel: A string kernel for SVM protein classification,” Proceedings of the Pacific Symposium on Biocomputing, ed. R.B. Altman, A.K. Dunker, L. Hunter, K. Lauerdale, and T.E. Klein, pp.566–575, World Scientific, 2002.

[10] M. Collins and N. Duffy, “Convolution kernels for natural language,” Advances in Neural Information Processing Systems 14, Cambridge, MA, MIT Press, 2002.

[11] H. Kashima and T. Koyanagi, “Kernels for semi-structured date,” Proceedings of the Nineteenth International Conference on Machine Learning, San Francisco, CA, pp.291–298, Morgan Kaufmann, 2002.

[12] H. Kashima, K. Tsuda, and A. Inokuchi, “Marginalized kernels between labeled graphs,” Proceedings of the Twentieth International Conference on Machine Learning, San Francisco, CA, Morgan Kaufmann, 2003.

[13] T. Gärtner, P. Flach, and S. Wrobel, “On graph kernels: Hardness results and efficient alternatives,” Proceedings of the Sixteenth Annual Conference on Computational Learning Theory, 2003.

[14] T. Gärtner, “A survey of kernels for structured data,” SIGKDD Explorations, vol.5, no.1, pp.S268–S275, 2003.

[15] H. Kashima and Y. Tsuboi, “Kernel-based discriminative learning algorithms for labeling sequences, trees, and graphs,” Proceedings of the Twenty-First International Conference on Machine Learning, 2004.

[16] Y. Altun, M. Johnson, and T. Hofmann, “Investigating loss functions and optimization methods for discriminative learning of label sequences,” Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2003.

[17] M. Collins, “Discriminative reranking for natural language parsing,” Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, pp.175–182, Morgan Kaufmann, 2000.

[18] H. Kanayama, K. Torisawa, Y. Mitsuishi, and J. Tsujii, “A hybrid Japanese parser with hand-crafted grammar and statistics,” Proceedings of the 18th International Conference on Computational Linguistics, pp.411–417, 2000.

(注4)：パーザの精度は 88%と報告されている [18].