

# 機械学習の数式処理への応用について

## Application of machine learning to computer algebra

小林宗広\*

MUNEHIRO KOBAYASHI

筑波大学大学院数理物質科学研究科

GRADUATE SCHOOL OF PURE AND APPLIED SCIENCES, UNIVERSITY OF TSUKUBA

### Abstract

Huang et al. published the paper of applying machine learning to computer algebra [1]. In this article, we see a brief summary of their paper. Also, introductory ideas of support vector machines are described as a related topic. Tree kernels appear to be applicable to training support vector machines on data of RCF formulas. We introduce the definition and an efficient way to calculate the values of tree kernels.

## 1 はじめに

Huang らにより機械学習を数式処理に応用する実験が報告された [1]. 本稿は Huang らの実験を紹介するとともに, 関連事項に関する解説を与えるサーベイレポートである. 関連事項として, Huang らの実験で利用された機械学習のアルゴリズムであるサポートベクターマシンに関する基本的な解説を与える. また, 木構造を持つデータの分類問題をサポートベクターマシンによって学習するために用いることができる木カーネルについても紹介する. 論理式や多項式には構成に関する木構造を自然に導入することができるため, 木カーネルをサポートベクターマシンによる論理式の分類問題の学習に利用できる.

2 章では Huang らの実験について紹介を行う [1]. 3 章でサポートベクターマシンに関する基本的事項について解説する. 3 章の内容は主に [5], [6] に基づく. 4 章では木カーネルを紹介する. 4 章をまとめるにあたり, 特に [3], [4], [7] を参照した.

## 2 Huang らの実験

Huang らにより効率的な Cylindrical algebraic decomposition (CAD) の変数順序を推定するために機械学習を用いた実験がなされた. ここではその報告 [1] を引用する.

### 2.1 実験概要

**目的** 論理式に CAD を適用するにあたり, 生成する細胞の数が最も少ない変数順序を推定するヒューリスティクスが 3 つ (Brown, sotd, ndrr) の内どれであるかを機械学習する.

---

\*munehiro-k@math.tsukuba.ac.jp

## ヒューリスティクス

Brown 入力された論理式から次の順番で変数を優先する：

- (1) 次数の低い変数
- (2) 入力に表れる各項の次数の総和が小さい変数
- (3) 入力において含まれる項の数が小さい変数

sotd 入力された論理式から各変数消去の順番に対し射影因子を全て構成し、各多項式に表れる項の全次数の和が最小となる順番を採用する。

ndrr 入力された論理式から各変数消去の順番に対し射影因子を全て構成し、各一変数多項式の異なる実根の総数が最小となる順番を採用する。

システム QEPCAD [9](CAD の実行), SVM-Light [10](機械学習の実行)

データ nlsat dataset [8] から 3 変数存在量化の論理式を 7001 個抽出して利用した。

- 流れ
- ・ 論理式を 11 次元の特性ベクトルに変換した。
  - ・ 論理式を QEPCAD を用いて各変数消去の順番で CAD を実行し、最も細胞数が少ない順序を推定したヒューリスティクスのラベルを特性ベクトルに付し訓練データとした。この際、CAD による量子消去のために必要な細胞数と論理式から量子を取り除き CAD のみを施した場合の細胞数の 2 系統の計測を行い、それぞれの訓練データを作成した。以下の実験はすべて各訓練データに対してそれぞれ行った。
  - ・ ラベル付きの特性ベクトルを training(3575), validation(1735), test(1721) に無作為に分割し、ベクトルの正規化を施した。
  - ・ SVM-Light を用いて 3 クラス分類を学習させた。学習は training と validation 上で RBF カーネルを用いて行った。
  - ・ 学習した分類器の性能を test 上で評価し、各ヒューリスティクスをそのまま用いた場合と比較対照した。

**結果** 機械学習の結果、CAD により生成される細胞が最も少ないヒューリスティクスを選ぶ場合が多かった。そのため、どのヒューリスティクスを単独で用いるよりも機械学習を用いた方が CAD により生成される細胞が少ない傾向にあった。

**課題** Huang らの実験は機械学習を数式処理に応用する初の大規模な研究であり、特性抽出やカーネルの選択の見直しなど、更なる研究の余地が残されている。

## 2.2 コメント

口頭発表時に Huang らの実験に対するコメントがあったため、それらをまとめておく。

- ・ 機械学習が有効であったと結論付けるには実験が不十分ではないか。別のデータセットや training, validation, test の別の分割の仕方に関しても同様の実験を繰り返し、統計を取るべきである。
- ・ ヒューリスティクスや学習結果の性能を評価する際には、CAD の細胞数ではなく実行時間を測定すべきである。3 変数では 3 回射影と持ち上げを行うが、最終的な CAD の細胞数が同じでも途中の分かれ方により実行時間に差が出るはずである。

- ・ 現実的範囲で量子化消去に CAD が適用できる変数の数は現状 5 個程度までであり、その順序も限られている。すべての変数順序による CAD を並列に動かせば十分ではないか。

### 3 サポートベクターマシン

#### 3.1 線形分離可能な 2 クラス分類の学習

サポートベクターマシンは与えられた訓練データを元に新しいデータの分類仮説を学習するためのアルゴリズムの一つである。

ここでは最も基本的な場合である線形分離可能な 2 クラス分類を学習するサポートベクターマシンを紹介する。訓練データを  $T = \{(x_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} \mid 1 \leq i \leq k\}$  とする。これらの訓練データ  $T$  を  $n$  次元超平面  $\langle x_i, w \rangle + b = 0$  によって肯定例  $P = \{x_i \mid y_i = 1\}$  及び否定例  $N = \{x_i \mid y_i = -1\}$  に分割することが学習の目的である。さらに、そのようにデータを分離する超平面のうち、肯定例  $P$  及び否定例  $N$  との距離の和が最大になるようなものを求めることとする。

この問題の数学的な定式化として次の最小化問題が得られる：

$$(OP) \begin{cases} \text{Minimize: } \|w\|^2, \\ \text{subject to: } y_i(\langle x_i, w \rangle + b) - 1 \geq 0 \quad (1 \leq i \leq k). \end{cases}$$

この問題を凸最適化問題の結果を用いて別問題に変換する。まず  $\bar{\lambda} = (\lambda_i)_{1 \leq i \leq k}$  として、ラグランジアン

$$\mathcal{L}(w, b, \bar{\lambda}) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^k \lambda_i (y_i(\langle x_i, w \rangle + b) - 1)$$

に対して

$$\frac{1}{2} \min_{w, b} \|w\|^2 = \min_{w, b} \max_{\bar{\lambda} \geq 0} \mathcal{L}(w, b, \bar{\lambda})$$

が成り立つ。ただし、 $\bar{\lambda} \geq 0$  は各  $1 \leq i \leq k$  に対し  $\lambda_i \geq 0$  のこととする。さらに、 $\|w\|^2$  が下に凸な関数であり、制約条件を満たす集合が凸集合であることから、この最適化問題は凸二次計画法問題となる。よって

$$\min_{w, b} \max_{\bar{\lambda} \geq 0} \mathcal{L}(w, b, \bar{\lambda}) = \max_{\bar{\lambda} \geq 0} \min_{w, b} \mathcal{L}(w, b, \bar{\lambda})$$

が成り立つ。 $\min_{w, b} \mathcal{L}(w, b, \bar{\lambda})$  に関しては  $w$  の各成分や  $b$  に関する偏微分が 0 という条件を考えることによりさらに変形することができ、最終的に次の  $\bar{\lambda}$  に関する最大化問題が得られる：

$$(DP) \begin{cases} \text{Maximize: } \sum_{i=1}^k \lambda_i - \frac{1}{2} \sum_{i, j=1}^k \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle, \\ \text{subject to: } \sum_{i=1}^k \lambda_i y_i = 0, \bar{\lambda} \geq 0. \end{cases}$$

元の最小化問題 (OP) の解は  $w = \sum_{i=1}^k \lambda_i y_i x_i$  で与えられる。 $b$  に関しては陰に定まるものの、陽に求めることができない。加えて Karush-Kuhn-Tucker 条件の一部である  $\lambda_i (y_i(\langle x_i, w \rangle + b) - 1) = 0$  ( $1 \leq i \leq k$ ) を考えることにより  $b$  も求めることができる。

(DP) の重要な特徴として、訓練データのベクトル  $x_i$  は内積の形のみで現れる点がある。(OP) は訓練データの次元 ( $n$ ) 個の変数と訓練データの数 ( $k$ ) 個の制約条件を持つ問題であった。一方 (DP) は、内積の値が計算できさえすれば、 $n$  によらず  $k$  個の変数、 $k$  個の制約条件を持つ問題である。(DP) のこの性質によりサポートベクターマシンの高速な動作及び非線形分離への拡張が実現される。

## 3.2 多クラス分類

サポートベクターマシンを用いて  $m$  クラス分類の学習を行うためには、訓練データから各クラスごとに分離超平面を求め、未知のデータを分類する先のクラスとして対応する分離超平面との距離が最も遠いものを選択する方法がある。

具体的には訓練データ  $T = \{(x_i, y_i) \in \mathbb{R}^n \times \{1, \dots, m\} \mid 1 \leq i \leq k\}$  に対し、 $j$  番目のクラス ( $1 \leq j \leq m$ ) に対応する訓練データ  $T_j \subset \mathbb{R}^n \times \{-1, 1\}$  を

$$(x_i, y_i) \in T_j \Leftrightarrow \begin{cases} (x_i, j) \in T & \text{if } y_i = 1 \\ (x_i, j) \notin T & \text{if } y_i = -1 \end{cases}$$

によって定義し、各  $T_j$  の 2 クラス分類をサポートベクターマシンにより学習、分離超平面  $\langle w_j, x \rangle + b_j = 0$  を得る。未知のデータ  $\tilde{x} \in \mathbb{R}^n$  の属するクラスの推定には各分離超平面からの距離

$$\frac{|\langle w_j, \tilde{x} \rangle + b_j|}{\|w_j\|}$$

を最大とするクラスを選択する。

## 4 Tree Kernel

### 4.1 カーネル法

サポートベクターマシンの非線形分離への拡張には、訓練データがよりよく分離されるような高次元のベクトル空間に非線形に埋め込み、その高次元空間で線形分離することを考える。その際、埋め込んだ先の高次元空間での内積を計算することができれば最小化問題 (OP) に対応する最大化問題 (DP) を解くことができ、非線形埋め込みの陽な表現までは知る必要がない。

カーネルとは埋め込み先の内積を表現する関数である。

**Definition 4.1.1.**  $X$  を集合とする。  $K: X \times X \rightarrow \mathbb{R}$  が正定値な対称写像であるとき、 $K$  をカーネルという。ただし、 $K$  が正定値とは  $N$  を任意の自然数として任意の  $x_1, \dots, x_N \in X$  及び  $c_1, \dots, c_N \in \mathbb{R}$  に対して  $\sum_{i,j} c_i c_j K(x_i, x_j) \geq 0$  が成り立つことをいう。

次の定理は実際にカーネルがあるヒルベルト空間での内積を表していることを主張している。定理はより一般に  $X$  が可分距離空間でも成り立つことが知られている [7]。

**Theorem 4.1.2 (Mercer 1909).**  $X$  を可算集合とする。以下が成り立つ：

- 任意のヒルベルト空間  $(H, \langle \cdot, \cdot \rangle)$  と埋め込み  $\varphi: X \rightarrow H$  に対し、 $\langle \varphi(*), \varphi(*) \rangle$  はカーネルとなる。
- 任意のカーネル  $K$  に対し、あるヒルベルト空間  $(H, \langle \cdot, \cdot \rangle)$  と埋め込み  $\varphi: X \rightarrow H$  が存在し、 $K(x, y) = \langle \varphi(x), \varphi(y) \rangle$  が成り立つ。

カーネル法を用いれば、ベクトル空間上の非線形分離問題だけでなく、一般の集合上でのクラス分類問題も解くことができる。一般の集合の分類問題をサポートベクターマシンを用いて学習するためには、その集合の各元の特性ベクトルを抽出するという形で一度ベクトル空間に陽に埋め込むことにより、ベクトル空間上の分類問題に還元する方法がある。一方でカーネル法を用いれば、ベクトル空間への埋め込みを陰に保ちつつサポートベクターマシンを動かすことができる。

## 4.2 木カーネル

木構造を持つデータを対象とする学習問題に利用されるカーネルとして木カーネルがある。論理式や多項式は構成に関する木構造を持つため、木カーネルを用いた学習が効果的である可能性がある。ここでは木カーネルの定義及び計算方法を紹介する。

まず取り扱う木のクラスを定義する。有限の半順序集合  $(T, <)$  で最小元を持ち、かつ任意の  $t \in T$  に対し  $\{s \in T \mid s < t\}$  が  $<$  に関して全順序集合となるものを有限木という。さらに、ラベル集合を  $\Sigma$  としてラベル  $\ell: T \rightarrow \Sigma$  が指定され、また  $T$  上の全順序  $<'$  が定義された有限木  $(T, <, \ell, <')$  を  $\Sigma$ -ラベル付き順序木という。ただし  $<'$  に関して兄弟の関係にある頂点の順序のみが興味の対象であるので、構造の同型を考えるにあたって兄弟の関係にない頂点の順序関係は単に無視し、同一視することとする。

$\Sigma$ -ラベル付き順序木  $(T, <, \ell, <')$  の部分集合  $S \subset T$  に対し、 $(S, <|_S, \ell|_S, <'|_S)$  がまた  $\Sigma$ -ラベル付き順序木となるとき、 $S$  が  $T$  の部分木であるという。

$\Sigma$ -ラベル付き順序木のなすクラス上のカーネルとして木カーネルが提案されている [3][4]。これは離散的な構造に対して定義できる一般的なカーネル関数である畳み込みカーネル [7] の木構造への応用である。

**Definition 4.2.3** ([3][4][7]). ラベル付き順序木  $T, T'$  に対し、木カーネル  $K$  を次のように定義する：

$$K(T, T') = \sum_{s \in S(T)} \sum_{s' \in S(T')} K^S(s, s').$$

ただし、ラベル付き順序木  $X$  に対し  $S(X)$  は  $X$  の部分木全体を表し、

$$K^S(s, s') = \begin{cases} 1 & \text{if } s = s' \text{ (ラベル付き順序木として)}. \\ 0 & \text{if } s \neq s'. \end{cases}$$

木カーネルは2つのラベル付き順序木に共通する部分木の数を数えるものである。[4]に木カーネルの値を計算量  $O(|T||T'|)$  で計算する方法が与えられているので引用する。

**Algorithm 4.2.4** ([4, 3.2 ラベル付き順序木カーネルの拡張]). ラベル付き順序木  $(T, <, \ell, <')$ ,  $(T', <', \ell', <'')$  と木カーネル  $K$  に対し、次が成り立つ：

$$K(T, T') = \sum_{v \in T} \sum_{v' \in T'} K^R(v, v').$$

ここで、

$$K^R(v, v') = \sum_{s \in S_v(T)} \sum_{s' \in S_{v'}(T')} K^S(s, s').$$

ただし、 $S_v(T)$  は  $v$  を根として持つ  $T$  の部分木全体とする。 $K^R$  の値は次の計算から求めることができる。 $v$  または  $v'$  が葉のとき

$$K^R(v, v') = \begin{cases} 1 & \text{if } \ell(v) = \ell'(v'), \\ 0 & \text{if } \ell(v) \neq \ell'(v'). \end{cases}$$

$v$  及び  $v'$  が葉でないとき

$$K^R(v, v') = \begin{cases} \bar{K}_{v, v'}^R(ch(v), ch(v')) & \text{if } \ell(v) = \ell'(v'), \\ 0 & \text{if } \ell(v) \neq \ell'(v'). \end{cases}$$

ただし,  $ch(v)$  は頂点  $v$  の子頂点の数である. このとき,  $\bar{K}_{v,v'}^R$  は次の再帰式により計算される:

$$\begin{aligned}\bar{K}_{v,v'}^R(i, j) &= \bar{K}_{v,v'}^R(i-1, j) + \bar{K}_{v,v'}^R(i, j-1) - \bar{K}_{v,v'}^R(i-1, j-1) \\ &\quad + \bar{K}_{v,v'}^R(i-1, j-1) \cdot K^R(ch(v, i), ch(v', j)), \\ \bar{K}_{v,v'}^R(i, 0) &= \bar{K}_{v,v'}^R(0, j) = 1\end{aligned}$$

ここで,  $ch(v, i)$  は頂点  $v$  の第  $i$  番目の子頂点を表す. ■

上記の計算法は  $K^R$  及び  $\bar{K}^R$  に動的計画法を用いることにより, 効率的に計算することができる.  $\bar{K}_{v,v'}^R(i, j)$  は  $v$  の  $i$  番目まで,  $v'$  の  $j$  番目までの子頂点に制限した場合の  $K^R(v, v')$  の値となっている. すなわち,  $T|_{v_{\leq i}} = \{v\} \cup \{t \in T \mid ch(v, k) < t \ (1 \leq k \leq i)\}$  とおけば

$$\bar{K}_{v,v'}^R(i, j) = \sum_{s \in S_v(T|_{v_{\leq i}})} \sum_{s' \in S_{v'}(T'|_{v'_{\leq j}})} K^S(s, s')$$

が成り立つ.

ラベル集合  $\Sigma$  を論理記号を含めた実閉体の言語とすれば, 実閉体の論理式は構成に関する木構造によって自然に  $\Sigma$ -ラベル付き順序木に解釈できる. これにより, 木カーネルを用いてサポートベクターマシンで論理式の種類問題を学習することができる.

## 5 まとめ

機械学習が CAD の変数順序を推定するヒューリスティクスの選択に効果的であるとする Huang らの実験を紹介し, その関連事項としてサポートベクターマシンの基礎的概念を解説した. Huang らの実験は機械学習を数式処理に応用する初の大規模な研究であり, 更なる研究の余地が残されている. それらの中で今回はカーネルの選択に着目した. 木カーネルを用いることにより, サポートベクターマシンで論理式の種類問題を学習することができる. 木カーネルは, 論理式から特性ベクトルを陽に抽出する作業を省き, 論理式の種類問題を陰に評価することを可能とする.

## 参 考 文 献

- [1] Huang, Zongyan, et al. "Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition." Intelligent Computer Mathematics, pp. 92-107. (Lecture Notes in Artificial Intelligence, 8543). Springer Berlin Heidelberg, 2014
- [2] 実験データが <http://www.cl.cam.ac.uk/~zh242/data/> に公開されている.
- [3] Collins, Michael, and Nigel Duffy. "Convolution kernels for natural language." Advances in neural information processing systems. 2001.
- [4] 鹿島久嗣, 坂本比呂志, 小柳光生. "木構造データに対するカーネル関数の設計と解析." 人工知能学会論文誌 = Transactions of the Japanese Society for Artificial Intelligence: AI 21 (2006): 113-121.
- [5] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2.2 (1998): 121-167.
- [6] Vapnik, Vladimir. "Universal learning technology: Support vector machines." NEC Journal of Advanced Technology 2.2 (2005): 137-144.

- [7] Haussler, David. "Convolution kernels on discrete structures." Vol. 646. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.
- [8] 論理式のデータセットが <http://cs.nyu.edu/~dejan/nonlinear/> に公開されている。
- [9] Brown, Christopher W. "QEPCAD B: a program for computing with semi-algebraic sets using CADs." ACM SIGSAM Bulletin 37.4 (2003): 97-108. ソースコードが <http://www.usna.edu/CS/~qepcad/B/QEPCAD.html> より配布されている。
- [10] Joachims, Thorsten. "Making large-Scale SVM Learning Practical." Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999. バイナリ及びソースコードが <http://svmlight.joachims.org/> より配布されている。