Boolean Kernel Classifier: Support Vector Machine を用いたブール関数の帰納学習

佐土原 健
産業技術総合研究所

概要: 本論文は, ブール関数の帰納学習アルゴリズム Boolean Kernel Classifier (BKC) を提案する. BKC は, Support Vector Machine (SVM) を学習エンジンとして用い, ブーリアンカーネルを2つの目的で利用する. 一つの目的は, SVM を任意の連言が張る特徴空間上で動作させるためであり, もう一つの目的は, 容量制御のためである. SVM を用いて, 特徴空間上の線形識別関数 f を学習した後, 長さが高々 k の連言が張る部分空間への射影 f^k を, ブーリアンカーネルを用いて計算し, 経験損失を増加させない最小の k を求める. そして, k により制限された部分空間で再学習を行い, より高い分類精度を持つ分類器を学習する. 本論文は, このような学習アルゴリズムが, C4.5 や Naive Bayes Classifier よりも優れた性能を有することをランダムに生成したブール関数の学習実験により示す.

Boolean Kernel Classifier: an inductive learning algorithm for Boolean functions using support vector machines

Ken Sadohara
National Institute of Advanced Industrial Science and Technology (AIST)

Abstract: This paper presents a new learning algorithm for Boolean functions called Boolean Kernel Classifier (BKC). BKC uses Support Vector Machines (SVMs) as learning engines, and it also uses Boolean kernels not only to run SVMs in the feature space consisting of all possible conjunctions, but also to control its capacity. After applying SVMs to learn discriminant functions f in the feature space, BKC uses Boolean kernels to compute the projections f^k of f onto the subspace consisting of conjunctions with length at most k. By evaluating the accuracy of f^k on training data, BKC find the smallest k such that f^k is as accurate as f. In the feature space constrained by k, BKC then relearns another f' expected to have lower error for unseen data. It is shown that BKC outperforms C4.5 and naive Bayes classifiers by an empirical study on learning of randomly generated Boolean functions.

1 はじめに

本論文は, ブーリアンカーネル関数と Support Vector Machine (SVM) [2, 10] を用いて, ブール関数の帰納学習を行うアルゴリズム Boolean Kernel Classifier (BKC) を提案する.

ブール関数の帰納学習に関しては, 計算論的学習理論 [1] において, 多くの研究が行われてきた. しかし, 一般のブール関数が, 現実的な計算資源の下で学習可能であるか否かは, 現在でも未解決のままである. 一方で, 連言の長さを高々 k に制限した選言標準形 (Disjunctive Normal Form; DNF) に関しては, 多項式時間学習可能性が証明されている. しかし, この証明で用いられているアルゴリズムは, 命題変数の数が d のとき, 訓練データの数が $O(d^k)$ 個必要になる等, 現実的な学習アルゴリズムとは言えない.

このような理論的な研究の一方で, 様々な応用領域ごとに多くの学習アルゴリズムがこれまでに提案されてきた. 例えば, データマイニングの分野においては, データが離散属性値のベクトルとして記述できる場合, 分類学習問題は, 原理的にブール関数の帰納学習問題に帰着できる. そして, このような問題に最も良く用いられているのが, C4.5 [8] や Naive Bayes Classifier (NBC) [6] である. しかし, これらの学習アルゴリズムに対しては, 強すぎる言語バイアスや過学習の問題が指摘されている [7, 3, 4].

本論文では, SVM を学習エンジンとして利用する, 新たなブール関数の学習アルゴリズム BKC を提案する. SVM は, 与えられた基本属性から構成される非常に多くの特徴が張る高次元特徴空間における効率の良い学習や, 統計的学習理論に基づく過学習の抑制等の利点を持っている. このような利点をブール関数の帰納学習に活かすために, ブール関数の学習に特化した SVM を提案し, その有効性を計算機実験によって示した [9]. このアルゴリズムでは, 与えられた命題変数から構成可能な全ての連言が張る特徴空間を用いる. 任意のブール関数は DNF 式で記述できるので, このような特徴空間上で, 常に線形分離可能となるからである. そして, この空間の線形識別関数を SVM を用いて高速に学習するために, 特徴空間の内積を $O(d)$ の計算量で計算できる DNF カーネル関数を提案した. このことは, 特徴空間を張る 3^d-1 個の特徴の分類に対する寄与度を, 多項式時間で推定できることを意味しており, 与えられた基本属性の貢献度しか推定しない C4.5 や NBC と比べて, 分類精度の向上が期待できる.

しかしながら, DNF カーネルを用いた SVM は, 命題変数の数 d が増加するとき, 著しく分類精度が

劣化することが明らかになった．この現象は，訓練データの数が少ないとき，連言の長さが長いほど，分類に対する寄与度の推定の精度が悪くなることに起因すると考えられる．従って，連言の長さを適切に制御することで，分類に寄与しない連言の影響を取り除くような，仮説空間の容量制御法が必要になる．もしも，何らかの方法で，連言の長さが高々 k $(k \leq d)$ で十分であるとわかれば，k により制限された部分空間の内積を計算するカーネル関数として，k-DNF カーネル関数が知られているので，この部分空間で SVM を動作させることができる．後は，k をいかに決めるかが問題となるが，本研究では，k の決定にも k-DNF カーネルを用いる．k-DNF カーネルを用いることで，学習により得られた識別関数 f に対して，長さ高々 k の連言が張る部分空間への射影 f^k を高速に計算できる．従って，このような射影 f^k を用いて訓練データを再評価することで，経験損失が増加しないような最小の k を求めることができる．

本論文では，上述したような，DNF カーネルや k-DNF カーネルといった一連のブーリアンカーネル関数を用いた SVM を学習エンジンとして用いると同時に，k-DNF カーネルを用いた容量制御機構を組み込んだブール関数帰納学習アルゴリズム BKC を提案し，その有効性をランダムに生成したブール関数の学習実験を通して明らかにする．

2 Support Vector Machines

SVM は，入力空間で非線形識別関数を学習する代わりに，特徴空間上で線形識別関数を学習する．入力空間 X 上の訓練データ $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ は，特徴写像 ϕ により $\phi(S) = \{(\phi(\boldsymbol{x}_i), y_i)\}_{i=1}^n = \{(\boldsymbol{z}_i, y_i)\}_{i=1}^n$ のように特徴空間に写像される．この特徴空間上で，正例 $\{\boldsymbol{z}_i \mid y_i = 1\}$ と負例 $\{\boldsymbol{z}_i \mid y_i = -1\}$ を分離する超平面 $f(\boldsymbol{z}) = \langle \boldsymbol{w} \cdot \boldsymbol{z} \rangle + b = 0$ を学習する．

特徴空間において，任意の超平面 $f(\boldsymbol{z}) = 0$ に最も近い点 $\boldsymbol{z}^*$ までの距離 $\frac{|f(\boldsymbol{z}^*)|}{\|\boldsymbol{w}\|}$ を，その超平面のマージンと呼ぶ．ただし，$\|\boldsymbol{w}\| = \langle \boldsymbol{w} \cdot \boldsymbol{w} \rangle^{\frac{1}{2}}$ とする．SVM は，$|f(\boldsymbol{z}^*)| = 1$ の正規化の下で，正例と負例を分離できて，しかも最も大きなマージンを持つ最適分離超平面を学習する．最適分離超平面の選択は，期待損失最小化の観点から正当化され [2, 定理 4.18]，未知のデータに対するエラーの上界を最小化することに相当している．結局 SVM の学習は以下のような凸二次計画問題として定式化できる．

最小化: $\|\boldsymbol{w}\|^2$, 制約条件: $y_i f(\boldsymbol{z}_i) \geq 1, (1 \leq i \leq n)$

凸二次計画問題であるので大域的最適解への収束が保証された最適化アルゴリズムが存在する．

詳細は省略するが，上述の二次計画問題は，以下のような双対問題に変換される．

最大化: $\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \boldsymbol{z}_i \cdot \boldsymbol{z}_j \rangle$

制約条件: $\alpha_i \geq 0 \ \ (1 \leq i \leq n), \ \sum_{i=1}^n \alpha_i y_i = 0$

双対問題の最適解 $\alpha_1^*, \ldots, \alpha_n^*$ に対して，最適分離超平面 $f(\boldsymbol{z}) = 0$ は以下のように書ける．

$$f(\boldsymbol{z}) = \sum_{i=1}^n \alpha_i^* y_i \langle \boldsymbol{z}_i \cdot \boldsymbol{z} \rangle + b^*$$

$$b^* = y_s - \sum_{i=1}^n \alpha_i^* y_i \langle \boldsymbol{z}_i \cdot \boldsymbol{z}_s \rangle \text{ for some } \alpha_s \neq 0$$

双対問題において，特徴空間の内積を計算する関数

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) \stackrel{\text{def}}{=} \langle \boldsymbol{z}_i \cdot \boldsymbol{z}_j \rangle = \langle \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) \rangle$$

をカーネル関数と呼ぶ．一般に特徴空間は非常に高次元の空間になるため，カーネル関数を，定義どおり ϕ を用いて計算することは現実的ではない．上記双対問題を実用的な時間で解くためには，特徴空間の次元に依存せずにカーネル関数を効率良く計算できる必要がある．

次節では，ブール関数の学習に SVM を適用するための特徴空間と，この空間における内積を効率良く計算できるカーネル関数について考察する．

3 ブーリアンカーネル

任意のブール関数は DNF 式で記述可能であるので、全ての連言が張る特徴空間のおいて，常に線形分離可能である [9]．このとき，$b = 0$ なる超平面だけ考えれば良いことも同時にわかる．例えば，2 変数 x_1 x_2 のブール関数に対しては，

$$x_1,\ x_2,\ 1 - x_1,\ 1 - x_2,\ x_1 x_2,\ x_1(1 - x_2),\ (1 - x_1)x_2,\ (1 - x_1)(1 - x_2)$$

の $3^2 - 1$ 次元の特徴空間を用いれば良い．このとき，$x_1 \vee \overline{x_2}$ に対しては，例えば，$f(x_1, x_2) = x_1 + (1 - x_1)(1 - x_2) - (1 - x_1)x_2$ のような識別関数によって線形分離できる．このような特徴空間への写像を ϕ_{DNF} と書くとき，ϕ_{DNF} の張る特徴空間のカーネル関数として，以下のものを用いることができる [9].

定義 1 (DNF カーネル)

$$K_{\text{DNF}}(\boldsymbol{u}, \boldsymbol{v}) \stackrel{\text{def}}{=} -1 + 2^{\text{sb}(\boldsymbol{u}, \boldsymbol{v})}$$

ここで，$\text{sb}(\boldsymbol{u}, \boldsymbol{v})$ は、$\boldsymbol{u}$ と $\boldsymbol{v}$ において同じ値を持つビットの数を表わす．

定理 1 $K_{\text{DNF}}(\boldsymbol{u}, \boldsymbol{v}) = \langle \phi_{\text{DNF}}(\boldsymbol{u}) \cdot \phi_{\text{DNF}}(\boldsymbol{v}) \rangle$.

命題 1 K_{DNF} の計算量は $O(d)$.

このように, 3^d-1 次元の特徴空間の内積を $O(d)$ で計算できるので, SVM を用いて特徴空間上の最適分離超平面を効率良く求めることが可能になる.

このような特徴空間は, 任意のブール関数に対して適用可能であるが, 期待損失を小さくするためには, データにあわせて効果的に容量制御を行なわねばならない. その目的で BKC は, 上記特徴空間の以下のような部分空間を用いる. ϕ_{DNF} が張る特徴空間に対して, 長さが高々 k の連言が張る部分空間を考え, この空間への特徴写像を ϕ^k_{DNF} と表わす. パラメタ k で制限されたこの特徴空間に対しては, 以下のようなカーネル関数を用いることができる [5].

定義 2 (k-DNF カーネル)

$$K^k_{\mathrm{DNF}}(\boldsymbol{u},\boldsymbol{v}) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{k}\binom{\mathrm{sb}(\boldsymbol{u},\boldsymbol{v})}{i}$$

定理 2 $\quad K^k_{\mathrm{DNF}}(\boldsymbol{u},\boldsymbol{v}) = \left\langle \phi^k_{\mathrm{DNF}}(\boldsymbol{u})\cdot\phi^k_{\mathrm{DNF}}(\boldsymbol{v})\right\rangle$.

定義からわかるように, DNF カーネルは, $k=d$ とした特別な k-DNF カーネルであることがわかる. K^k_{DNF} の計算量に関して, $\binom{s}{i}=\binom{s-1}{i}+\binom{s-1}{i-1}$ に注意すれば, 以下の命題が成立する.

命題 2 K^k_{DNF} の計算量は $O(dk)$.

実際には, 事前に $d\times d$ のテーブルを作っておけば, 無駄な計算を省くことができ, 学習時には定数時間でカーネル関数値を参照できる.

この節の議論は, 否定を含まない連言が張る特徴空間に対しても, そのまま成立する. その場合は, $\boldsymbol{u}$ と $\boldsymbol{v}$ において, ともに値 1 を持つビットの数を表わす $\mathrm{sp}(\boldsymbol{u},\boldsymbol{v})$ を $\mathrm{sb}(\boldsymbol{u},\boldsymbol{v})$ の代わりに用いれば良い. このようにして, DNF カーネルに対応する単調 DNF カーネル, k-DNF カーネルに対応する, 単調 k-DNF カーネルが得られる. これらのカーネル関数を総称してブーリアンカーネル関数と呼ぶ.

4 Boolean Kernel Classifier

BKC の特徴は k-DNF カーネルを用いた容量制御にある. k-DNF カーネルで張られた特徴空間上の関数の集合を S_k と表わせば, $S_1\subseteq S_2\subseteq\cdots\subseteq S_d$ である. 構造的損失最小化 (Structural Risk Minimization) [10] に従えば, 関数 $f_i\in S_i$, $f_j\in S_j$ $(i<j)$ が同じ経験損失を持つとき, f_i の方がより小さな期待損失を持つことが期待できる.

そこで, 学習の結果, $f_j\in S_j$ なる識別関数が得られ, f_j の経験損失が R であるとき, BKC は, $f_i\in S_i$, $(i\le j)$ で経験損失が R であるものが存在するかどうかを調べる.

この目的のために, BKC は, f_j を, 高々長さ i の連言が張る特徴空間に射影して得られる関数 f^i_j を用いて訓練データを評価し, 経験損失が R と同じであるかどうかを調べる.

以下の定理は, 上記の射影が, k-DNF カーネルを用いて高速に計算できることを意味している.

定理 3 $f(\boldsymbol{x})=\sum_{j=1}^n y_j\alpha_j K^\ell_{\mathrm{DNF}}(\boldsymbol{x}_j,\boldsymbol{x})$, $k\le\ell\le d$ とするとき, $f^k(\boldsymbol{x})=\sum_{j=1}^n y_j\alpha_j K^k_{\mathrm{DNF}}(\boldsymbol{x}_j,\boldsymbol{x})$.

このように, k-DNF カーネルを, いわばローパスフィルタのように用いて, 識別関数の $O(d^\ell)$ 個の項の中から, 長さが高々 k の連言に対応する成分だけを効率良く抽出する点が BKC の特徴の一つである.

以下は, BKC の学習手続きの概略である.

1. $k\stackrel{\mathrm{def}}{=}d$.
2. k-DNF カーネルと SVM を用いて識別関数 f を学習する.
3. 訓練データを f^i, $(1\le i\le k)$ を用いて評価し, 分類精度が劣化しない最小の i を求める.
4. もし, $i=k$ ならば終了し, そうでなければ, $k\stackrel{\mathrm{def}}{=}i$ とし, 2 に戻る.

5 実験

BKC の性能を検証するために, ランダムに生成したブール関数を学習する実験を行い, NBC, C4.5 との比較を行ってみた. 実験の詳細は以下の通りである.

100 個のブール関数をランダムに生成する. 各ブール関数ごとに, n 個の訓練データと 2000 個のテストデータを一様分布の下でランダムに生成し, テストデータに対する分類精度を測定する. そして, 100 回の測定の平均をとる.

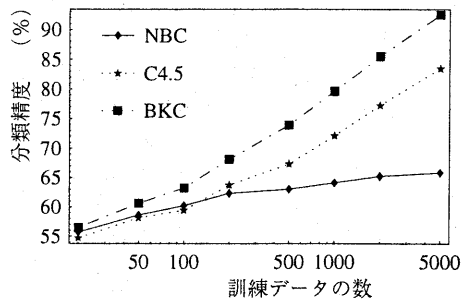
ブール関数の生成は, d 個の変数からなる DNF 式を以下のように生成することで行う. 各変数は, 連言のリテラルとして $\frac{\ell}{d}$ の確率で選ばれ, さらに $\frac{1}{2}$ の確率で負リテラルとなる. 従って, 連言の長さは, 平均 ℓ の二項分布 $B(d,\frac{\ell}{d})$ に従う. 連言の数は, 正例と負例がほぼ同じ割合で生成されるように $2^{\ell-2}$ とした.

グラフ 1 は, $d=16$, $\ell=8$ の時に, n を変化させた場合の分類精度の変化を示している.

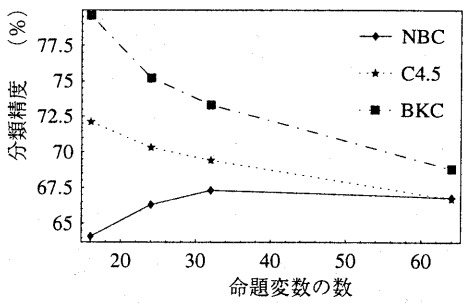
次に, グラフ 2 は, $\ell=8$, $n=1000$ の時に, d を変化させた場合の分類精度の変化を示している.

最後に, グラフ 3 は, $d=16$, $n=1000$ の時に, ℓ を変化させた場合の分類精度の変化を示している. なお, この実験に限り, DNF 式の連言の長さは, 平均 ℓ ではなく, 正確に ℓ とし, 連言の数は $2^{\ell-1}$ とした.

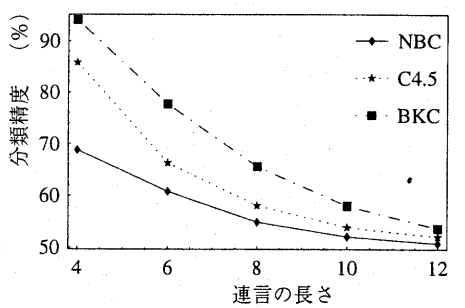
以上のように, 訓練データの数, 命題変数の数, DNF 式の複雑さの 3 つのパラメータを変化させ, BKC , NBC, C4.5 の分類精度を比較してみた結果, いずれの場合にも BKC の優位性を確認できた.



グラフ 1: 訓練データの数に対する分類精度の変化



グラフ 2: 変数の数に対する分類精度の変化



グラフ 3: DNF 式の複雑さに対する分類精度の変化

6 おわりに

本稿では, ブーリアンカーネルを用いた SVM を学習エンジンとして用いると同時に, ブーリアンカーネルを容量制御に用いるブール関数の帰納学習アルゴリズム Boolean Kernel Classifier (BKC) を提案した. また, ランダムに生成したブール関数の学習実験を通して, BKC が C4.5 や NBC より優れた性能を持つことを示した.

この実験により BKC は, C4.5 よりも, 高い分類精度を持つことがわかった. しかし, C4.5 は決定木を学習するので, 学習によって得られた分類器が人間にとってより理解しやすいという利点を持っている. これに対して, BKC により得られる分類器は, 任意の連言の分類に対する寄与度を内包しているものの, そのままでは, 人間にとって理解することが難しい. データマイニングにおいては, 分類精度と並んで, 分類器の可読性も重要な要件であるので, 連言の寄与度に基づいて, 分類の根拠をユーザが理解しやすい形で提示する手法の開発が今後の課題である.

参考文献

[1] 西野 哲朗 有川 節夫. 学習における計算論的アプローチ. 情報処理, 32(3):217–225, 1991.

[2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge Press, 2000.

[3] P. Domingos. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[4] P. Domingos. The role of occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3:409–425, 1999.

[5] R. Khardon, D. Roth, and R. Servedio. Efficiency versus convergence of boolean kernels for on-line learning algorithms. Technical Report UIUCDCS-R-2001-2233.

[6] T.M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[7] J.R. Quinlan. An empirical comparison of genetic and decision-tree classifiers. In *Proc. of ICML*, pages 135–141, 1988.

[8] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[9] Ken Sadohara. Learning of Boolean functions using support vector machines. LNAI 2225, pages 106–118. Springer, 2001.

[10] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.