解　説

ロボティクスに使えるやさしいSupport Vector Machine

Easy Support Vector Machine for Robotics

吉　川　雅　博*　*産業技術総合研究所
Masahiro Yoshikawa*　*National Institute of Advanced Industrial Science and Technology (AIST)

1. はじめに

Support Vector Machine（以下，SVM）は，Vapnik らによって提案されたパターン認識手法である [1]．その登場から約 15 年が経ち，コンピュータビジョン，自然言語処理，バイオインフォマティクスなど幅広い分野で用いられ，汎用的なパターン認識手法としての地位を確立している．SVM は識別性能の高さもさることながら，手軽に使えるオープンソースのライブラリが充実していることが大きな強みである．本稿ではロボット研究にこれから SVM を使ってみようと考えている方を対象に，SVM の原理とライブラリの使用方法を説明し，SVM の応用事例として筋電位から手の動作意図を推定する研究について紹介する．

2. SVM の原理

2.1 まずはパターン認識の基本から

パターン認識のわかりやすい説明として，前田のキノコの例を拝借する [2]．秋の行楽にキノコ採りに来たとしよう．あなたは毒キノコと食べられるキノコを見分けてキノコを採らなければならない．あなたの手元にあるのは，食べられるキノコ 15 本，毒キノコ 15 本のサンプル．あなたは山で見つけた得体の知れないキノコが，食べられるキノコなのか，毒キノコなのかをサンプルだけを頼りに見分ける方法をみつけなければならない．

キノコが持っている様々な特徴のうち，色，重さ，柄の長さ，斑点の数，などの定量的に扱えるものを特徴量と呼ぶ．ここで d 個の特徴量があるとすると，キノコの特徴は d 次元ベクトル $\boldsymbol{x}$ で表せる．この $\boldsymbol{x}$ をパターン，もしくは特徴ベクトルと呼び，食べられるキノコと毒キノコのサンプルを学習データと呼ぶ．ここでは，簡単に特徴量を「柄の長さ」と「傘の体積」に絞って考える．食べられるキノコと毒キノコをグラフにプロットすると図 1 になったとする．特徴量を座標軸にとった空間を特徴空間と呼ぶ．食べられるキノコや毒キノコといったカテゴリをクラスと呼び，ここではそれぞれのクラスを X_1，X_2 とする．

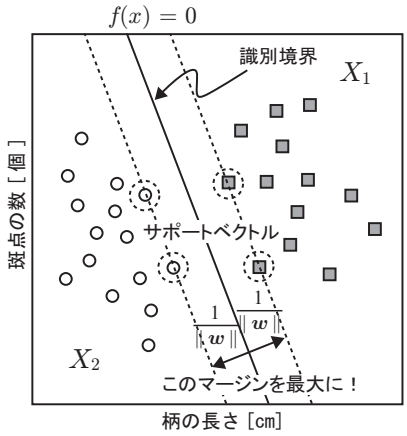


図 1　サポートベクトル，マージン，識別境界の関係

さて，未知のキノコのパターン $\boldsymbol{x}$ が属するクラスを判定するためには，図 1 の特徴空間に 2 つのクラスを分ける境界がわかればよい．SVM ではこの境界を求めるために，マージン最大化という明確な基準を用いる．図 1 のように，学習データの中で他のクラスから最も近い位置にあるサンプル（サポートベクトル）を基準として，それらのユークリッド距離が最大になる位置に境界を決定する．このマージン最大化基準は「真ん中を通す」という人間の直感に近く，未知のパターンへの識別能力すなわち汎化能力を高めるのに貢献している．

2.2 線形 SVM

それではこのマージン最大化基準を基づき，線形の識別境界を求める方法について説明する．未知のパターン $\boldsymbol{x}$ を識別するための線形な識別関数を

$$f(\boldsymbol{x}) = \boldsymbol{w}^t\boldsymbol{x} + w_0 \qquad (1)$$

と定義する．$\boldsymbol{x}$ は未知パターン，d 次元ベクトル $\boldsymbol{w}$ と w_0 は識別関数を構成するパラメータである．すべての学習サンプル $\boldsymbol{x}_i (i = 1, 2, \ldots, n)$ に対し，次の条件を満たすように $\boldsymbol{w}$ と w_0 を調整できれば，

原稿受付
キーワード：Support Vector Machine, SVM
*〒 305-8568 つくば市梅園 1-1-1
*Tsukuba-shi, Ibaraki

$$f(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x}_i + w_0 \begin{cases} \geq 1 & \text{if } \boldsymbol{x}_i \in X_1 \\ \leq -1 & \text{if } \boldsymbol{x}_i \in X_2 \end{cases} \quad (2)$$

未知のパターン $\boldsymbol{x}$ が与えられた時，$f(\boldsymbol{x})$ の符号の正負によっていずれかのクラスに識別できる．$\boldsymbol{w}$ と w_0 を求める過程を学習（もしくは訓練）と呼ぶ．$\boldsymbol{x}_i$ に対応するクラスを y_i で表し，次のように定義する．

$$y_i = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \in X_1 \\ -1 & \text{if } \boldsymbol{x}_i \in X_2 \end{cases} \quad (3)$$

式 (3) を使うと式 (2) は，一つの式で表現できる．

$$y_i(\boldsymbol{w}^t \boldsymbol{x}_i + w_0) - 1 \geq 0 \quad (4)$$

図 1 に示すように，サポートベクトルから $f(\boldsymbol{x}) = 0$ までの距離は $1/||\boldsymbol{w}||$ となるので，マージン最大化基準を満たすためには，$2/||\boldsymbol{w}||$ が最大になればよい．微分しやすくするために $||\boldsymbol{w}||$ を 2 乗すると，式 (4) の条件の下で，

$$G(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w}||^2 \quad (5)$$

を最小化する $\boldsymbol{w}$ と w_0 を求める問題になる．

ここでラグランジュの未定乗数 $\boldsymbol{\lambda}(\lambda_i \geq 0, i = 1, 2, \ldots, n)$ を導入し，式 (4) と式 (5) から成る制約条件付きの極値問題を関数 L を最小化する問題に置き換える．

$$L(\boldsymbol{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \lambda_i [y_i(\boldsymbol{w}^t \boldsymbol{x}_i + w_0) - 1] \quad (6)$$

L が極値を持つために，$\boldsymbol{w}$ と w_0 で偏微分して 0 とおくと，

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \lambda_i y_i \boldsymbol{x}_i = 0 \quad (7)$$

$$\frac{\partial L}{\partial w_0} = \sum_{i=1}^{n} \lambda_i y_i = 0 \quad (8)$$

これを式 (6) に代入して整理すると，L を最大化する λ_i を求める問題に落ち着く．

$$\text{Maximize} \quad L(\lambda_i) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^t \boldsymbol{x}_j \quad (9)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \lambda_i y_i = 0, \ \lambda_i \geq 0$$

これは凸 2 次計画問題であり，必ず大局的最適解を持つ．そのため，λ_i は最急降下法などで求めることが可能である．最終的に SVM の線形な識別関数は求めた λ_i^* を用いて，

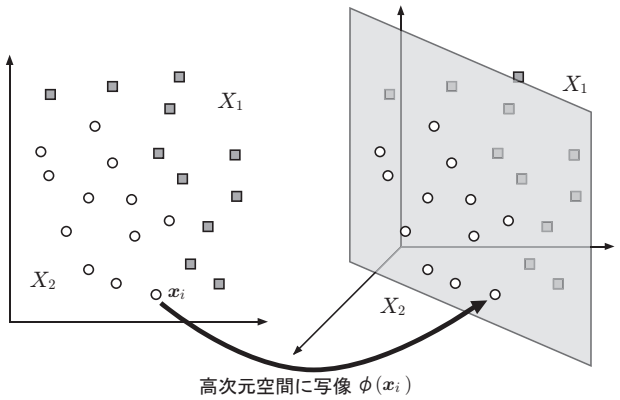


図 2 非線形 SVM は高次元への写像によって線形分離する

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i^* y_i (\boldsymbol{x}_i^t \boldsymbol{x}) + w_0 \quad (10)$$

となる．多くの場合 $\lambda_i = 0$ となり，残った λ_i^* に対応する学習サンプル $\boldsymbol{x}_i$ がサポートベクトルとなる．

ここまで説明してきた SVM はハードマージン線形 SVM と呼ばれ，2 つのクラスが線形な識別境界で完全に分離できる（線形分離可能な）場合に適用できる．現実のデータにはノイズが含まれており，線形分離不可能な場合も多いので一部の学習サンプルがマージン内に食い込んだり誤識別したりすることを許容することにより線形 SVM を適用可能とする．これをソフトマージンと呼ぶ．

2.3 非線形 SVM

次に，非線形な識別境界を構成可能な非線形 SVM について説明する．非線形 SVM では図 2 のように，線形分離不可能な学習データを，高次元の特徴空間に写像し，写像先の特徴空間において線形分離可能な識別境界を求める．ここで，d 次元の特徴ベクトル $\boldsymbol{x}$ を，d' 次元特徴空間に写像する関数を $\boldsymbol{\phi}(\boldsymbol{x})$ とおくと，非線形 SVM の識別関数は，

$$f(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{\phi}(\boldsymbol{x}) + w_0 \quad (11)$$

と表せる．このとき，線形 SVM において式 (9) で表した最大化問題は，非線形 SVM では次式となる．

$$\text{Maximize} \quad \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \boldsymbol{\phi}(\boldsymbol{x}_i)^t \boldsymbol{\phi}(\boldsymbol{x}_j) \quad (12)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \lambda_i y_i = 0, \ 0 \leq \lambda_i \leq C$$

ここで C は前項で触れたソフトマージンを構成するために，学習前に予め決めておくパラメータ（超パラメータ）であり，C を大きくするほどハードマージンに近づく．

ここで式 (12) の内積計算 $\boldsymbol{\phi}(\boldsymbol{x}_i)^t \boldsymbol{\phi}(\boldsymbol{x}_j)$ をそのまま計算しようとすると，高次元のベクトル演算が発生し計算量が

膨大になってしまう．そこで，$\boldsymbol{\phi}(\boldsymbol{x}_i)^t\boldsymbol{\phi}(\boldsymbol{x}_j)$ の演算を避けるために，計算が容易なカーネル関数 $K(\boldsymbol{x}_i, \boldsymbol{x})$ と呼ばれる関数で置き換える．

$$\text{Maximize} \quad \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{n}\lambda_i\lambda_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}) \quad (13)$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\lambda_i y_i = 0,\ 0 \le \lambda_i \le C$$

線形 SVM の場合と同様に λ_i^* を求めると，最終的に非線形 SVM の識別関数は，

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}\lambda_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + w_0 \quad (14)$$

となる．内積計算をカーネル関数で置き換える手法は，カーネルトリックと呼ばれ SVM における重要な工夫の一つである．カーネル関数は多数提案されているが，よく使われるものとして，次式で表す動径基底関数（RBF:Radial Basis Function）カーネルがある．

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp\left(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}||^2\right) \quad (15)$$

ここで，$\gamma(>0)$ は超パラメータであり，大きい値をとるほど学習データを明確に分離する複雑な識別境界を構成するようになるので，過学習（オーバーフィッティング）に注意が必要である．RBF カーネルを用いた非線形 SVM は，基本的にはこの γ と先に述べた C の 2 つの超パラメータを学習の前に決めておくだけでよく，超パラメータの多いニューラルネットよりも気楽である．

ここまでの説明を基に，SVM の特長をまとめる．

(1) マージン最大化基準に基づく識別関数により汎化能力が高い．
(2) 大域的最適解に収束するので，局所的最適解に陥る心配がない．
(3) λ_i の多くは 0 となり，少数のサポートベクトルで識別関数を表せる．
(4) 線形分離不可能な学習データに対しても，カーネルトリックにより少ない計算量で学習できる．
(5) RBF カーネルを用いた場合，超パラメータは C と γ のみとなり求めやすい．

3. SVM のライブラリの使用方法

SVM の普及には実用的で手軽に試せるオープンソースのライブラリの貢献が大きい．様々なライブラリが存在しているが，使い勝手がよいことと，複数のプログラム言語の実装が存在することから LIBSVM [3] を紹介する．2011 年 4 月 1 日に最新版の Version3.1 がリリースされている．具体的な使用方法を説明する前に，LIBSVM が複数のクラスを識別する際の処理方法について説明する．これまで説明したように SVM は原理的に 2 クラスを識別する手法であった．LIBSVM では複数のクラスを識別するために 1 対 1 比較法を用いて対応している．この方法では，N 個のクラスの全ての組み合わせ，すなわち $N(N+1)/2$ 個の識別関数を構成し，それぞれの識別関数を用いて未知パターンの識別を複数回行う．全ての識別結果を集計し，最も多く識別されたクラスを最終的な識別結果としている．

3.1 環境構築と学習・テストデータの準備

Windows 上での作業を前提として説明を進める．公式サイト（`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`）より，libsvm-3.1.zip をダウンロードし，解凍後作成される”windows”フォルダ内で作業を行う．”windows”フォルダには各種実行形式のツール群がある．LIBSVM data sets のページより，サンプルデータ svmguide1, svmguide1.t をダウンロードし，それぞれ sample_tain（学習用），sample_test（テスト用）にリネームし，”windows”フォルダに格納する．学習データ sample_tain を確認すると，各行にクラスラベル，特徴量 1，特徴量 2，特徴量 3，特徴量 4 の順に列記されているのがわかる．

```
0 1:4.2362 2:2.1982 3:3.5037 4:9.7521
0 1:1.6570 2:8.0663 3:2.1260 4:6.8668
...
1 1:7.7948 2:1.9367 3:1.5848 4:1.2226
1 1:5.0243 2:3.1211 3:1.6666 4:1.7998
...
```

3.2 特徴量のスケーリング

各特徴量のレンジが大きく異なると識別率が低下する．まずは，svm-scale.exe により学習データとテストデータの各特徴量を-1 から 1 のレンジに収まるようにスケーリングする．

```
svm-scale sample_train > sample_train_s
svm-scale sample_test > sample_test_s
```

3.3 学習

学習データを基に学習を行うプログラムは，svm-train.exe である．

```
svm-train [option] 学習データ [モデルファイル]
```

オプションを指定しない場合は，ソフトマージン非線形 SVM，RBF カーネルで学習が行われる．-t のオプションを指定することでカーネルを変更することができるが，まずは汎用的に使用できる RBF カーネルをお勧めする．

-v n のオプションを追加すると，n 重交差検定を行う．n 重交差検定は，学習したモデル（識別関数）が未知のテストデータに対しどの程度の識別性能を発揮するか学習データを基に推定する．学習データをランダムに n 分割し，分割したうちの 1 つをテストデータとして，残りのデータで

学習し識別を行う．これを n 回を繰り返し識別率の平均値を求めることで，実際のデータに対する識別率を推定する．

RBF カーネルを使用する場合，学習を実行する際に考慮が必要なオプションは，-c と-g であり，それぞれ式 (13) の C と式 (15) の γ に対応する．これは学習を行う前に適切な値を決定しておく必要がある．この方法としては，C と γ の組み合わせを用意し（例えば，$C = 2^{-5}, 2^{-4}, \ldots, 2^{5}$, $\gamma = 2^{-15}, 2^{-14}, \ldots, 2^{10}$），各組み合わせで交差検定を行い，最も高い識別率を示す組み合わせを選択する格子探索（グリッドサーチ）がある．この処理を自動で行う python スクリプト grid.py は，”tools”フォルダにある．

例えば，交差検定で $C = 2$ と $\gamma = 2$ が決定したとして，以下を実行すると model に学習したモデル（識別関数のパラメータなど）が保存される．

```
svm-train -c 2 -g 2 sample_train_s model
```

3.4 識別

テストデータの識別を行うプログラムは，svm-predict.exe である．

```
svm-predict テストデータ モデルファイル 出力結果
```

以下を実行すると識別率が表示され，output に sample_test_s を識別した結果が出力される．

```
svm-predict sample_test_s model output
Accuracy = 96.225% (3849/4000)
```

以上が簡単な使用方法であるが，自作の C/C++プログラムに組み込む場合は，svm.h をインクルードし，svm.cpp をリンクすれば良い．各関数の使用方法は README に詳しく記載されているが，基本的には svm_load_model で事前に学習したモデルファイルを読み込み，svm_predict で特徴ベクトルを識別するという流れになる．

4. SVM の応用事例

SVM は汎用的なパターン認識手法であるため，ロボットによるオブジェクトの認識，センサ情報に基づく自己位置の推定や周辺環境の状態推定など，実に様々な場面に適用可能である．最後に SVM の応用事例として，筆者らの研究を紹介する [4].

筋電義手は，前腕切断者が使用する電動義手であり，前腕に残存する筋の収縮時に発生する筋電位を制御信号とする．手の開閉のみを行える筋電義手がすでに普及しているが，さらに多自由度を制御可能な筋電義手の開発が期待されている．そこで，多自由度の筋電義手を制御するために，SVM を用いて前腕の筋電位から手の動作意図を推定する手法を提案した．本手法では前腕から 4 チャネル分の筋電位を計測し，各チャネルの筋電位から積分筋電位（1 次元）とケプストラム（4 次元）を特徴量として抽出し，計 16 次元の特徴ベクトルを得る．予め決められた順に動作を行うだけで，特徴ベクトルにクラスラベルを対応づけて学習データを作成し，超パラメータの決定，学習完了までを 30 秒程度で（Core2Duo 2.4GHz の場合）自動的に行えるシステムを構築した．

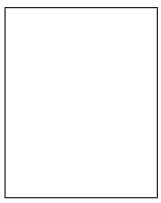
手首の屈曲，手首の伸展，指を閉じる，指を開く，前腕回内，前腕回外，中立動作の 7 動作の認識を行ったところ 8 名の被験者平均で 94.6%の動作識別率を得た．筋電位のパターンは疲労などにより経時変化し，時間と共に識別率が大きく低下するが，提案した手法は長時間に渡って高い識別率を維持できた．これらの結果は，従来のパターン認識手法を用いた場合よりも優れていた．また，62.5Hz のタイミングでリアルタイムに動作識別が可能であり，動作の意図に対して遅れがなく応答性がよいことが示された．

5. お わ り に

本稿では，ロボット研究にこれから SVM を使ってみようと考えている方を対象に，導入となる解説を行った．SVM は使い勝手の良いライブラリが充実しているため，あまり手間をかけずに優れたパターン認識手法を利用することができる．日本語の優れた解説も多数存在するので [2] [5]〜[11]，深く学びたい場合にはぜひご参照いただきたい．

参 考 文 献

[1] C. Cortes and V. Vapnik : “Support-Vector Networks, ” Machine Learning, Vol. 20, No. 3, pp. 273–297, 1995.
[2] 前田英作：“痛快！サポートベクトルマシン-古くて新しいパターン認識手法-”, 情報処理, Vol. 42, No. 7, pp. 676–683, 2001.
[3] C. C. Chang and C. J. Lin : LIBSVM: A Library for Support Vector Machines, 2001. www.csie.ntu.edu.tw/˜cjlin/libsvm.
[4] 吉川雅博, 三河正彦, 田中和世：“筋電位を利用したサポートベクターマシンによる手のリアルタイム動作識別, ” 信学論（D）, Vol. J92-D-II, No. 1, pp. 93–103, 2009.
[5] 前田英作：カーネル情報処理入門-非線形の魅惑-, 八木康史, 斉藤英雄（編）, コンピュータビジョン最先端ガイド 2 , pp. 91–131. アドコム・メディア, 2010.
[6] ビショップ C.M. , 元田浩, 栗田多喜夫, 樋口知之, 松本祐治, 松田昇（監訳）：パターン認識と機械学習 下, シュプリンガー, 2008.
[7] 荒木雅弘：フリーソフトでつくる音声認識システム, 森北出版, 2007.
[8] 小野田崇：サポートベクターマシン, オーム社, 2007.
[9] 渡辺澄夫, 萩原克幸, 赤穂昭太郎, 本村陽一, 福水健次, 岡田真人, 青柳美輝：学習システムの理 1 論と実践, 森北出版, 2005.
[10] N. Cristianini, J. Shawe-Taylor, 大北 剛（訳）：サポートベクターマシン入門, 共立出版, 2005.
[11] 麻生英樹, 津田宏治, 村田昇：パターン認識と学習の統計学-新しい概念と手法, 岩波書店, 2003.



吉川雅博（Masahiro Yoshikawa）

2010 年筑波大学大学院図書館情報メディア研究科博士課程修了．博士（情報学）．2010 年より（独）産業技術総合研究所知能システム研究部門特別研究員．生体信号を用いたヒューマンインタフェースとその福祉応用の研究に従事．（日本ロボット学会正会員）