2024年度

卒業論文

論文題目

Artificial Bee Colony アルゴリズムによるサポートベクトルマシンのハイパーパラメータ最適化

研究者

2131007　安達 拓真

指導教員

山口　智　　教授

2024年1月xx日

目 次

1　はじめに

内容 1

2　サポートベクトルマシン (SVM)

SVM は 1995 年に Cortes らによって提案された機械学習アルゴリズムである [[2]]．SVM は教師あり学習を用いる 2 値分類器であるが、多クラスへの拡張も可能である．SVM は，訓練サンプル集合からデータを分類する識別関数を学習するアルゴリズムである．この学習過程において，SVM は訓練データの中で識別関数に近いデータであるサポートベクトルを得る．そして，サポートベクトルと識別関数との距離，すなわちマージンを最大化するように識別関数を構築する．これにより，SVM は新しいデータを分類する際，最大限のマージンを持ってデータを分類できるようになり，未知のデータに対しても誤分類のリスクを最小限に抑えることができる．

2.1　ハードマージン SVM

n 個の N 次元教師データ $\boldsymbol{x}_i (i = 1, ..., n)$，がクラス K_1,K_2 の 2 値分類データであるとして，対応するクラスラベルをクラス K_1 の時に $y_i = 1$, クラス K_2 の時に $y_i = -1$, とするとデータが線形分離可能であれば，識別関数は (2.1) 式のように定義される．$\boldsymbol{w}$ は m 次元ベクトル, b は識別関数のパラメータである．

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b \tag{2.1}$$

すべての教師データ $\boldsymbol{x}_i (i = 1, ..., n)$ について (2.2) 式の条件を満たすように $\boldsymbol{w}$, b を調節する事が SVM の学習フェーズとなる．

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}_i + b = \begin{cases} \geq 1 & \boldsymbol{x}_i \in K_1 \\ \leq -1 & \boldsymbol{x}_i \in K_2 \end{cases} \tag{2.2}$$

このとき，(2.2) 式は (2.3) 式と等価になる．

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 \tag{2.3}$$

ここにサポートベクトルと超平面（線）の図

ここで図??にデータを分離する超平面 (2 次元では直線) を示す．分離超平面とそれに最も近いデータ (サポートベクトル) との距離をマージンと呼び，SVM はマージンを最大化するように分離超平面を求める．マージンを最大化することで，サポートベクトルから少しずれたデータが存在しても分類可能になり，モデルの汎化性能が向上する．次に，マージンを最大化する超平面を求める方法を導く．$\boldsymbol{x}_i$ から超平面への距離を d とすると，d は (2.4) 式のようになる．

$$d = \frac{|f(\boldsymbol{x_i})|}{\|\boldsymbol{w}\|} \tag{2.4}$$

よって，マージンを M とすると，マージン最大化は (2.5) 式の最適化問題によって表現できる

$$\max_{\boldsymbol{w},b} M, \ \ \text{subject to } \frac{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|} \geq M \ \ (i = 1, ..., n) \tag{2.5}$$

さらに，(2.5) 式の制約条件部分を M で割り，$\frac{\boldsymbol{w}}{M\|\boldsymbol{w}\|}$ と $\frac{b}{M\|\boldsymbol{w}\|}$ をそれぞれ $\tilde{w}$,$\tilde{b}$ とすると制約条件は (2.6) 式に書き換えられる．

$$y_i(\tilde{\boldsymbol{w}}^T\boldsymbol{x}_i + \tilde{b}) \geq 1 \tag{2.6}$$

(2.6) 式の等号が成立するデータ $\boldsymbol{x_i}$ は分離超平面と距離が最も近いデータ，すなわちサポートベクトルである．したがってマージン $\tilde{M}$ は (2.7) 式のようになる

$$\tilde{M} = \frac{1}{\|\tilde{\boldsymbol{w}}\|} \tag{2.7}$$

よって (2.5) 式は (2.8) 式で表すことができる．

$$\max_{\tilde{\boldsymbol{w}},\tilde{b}} \frac{1}{\|\tilde{\boldsymbol{w}}\|}, \ \ \text{subject to } y_i(\tilde{\boldsymbol{w}}^T\boldsymbol{x}_i + \tilde{b}) \geq 1 \ \ (i = 1, ..., n) \tag{2.8}$$

$\|\tilde{\boldsymbol{w}}\|$ は正であるため，$\frac{1}{\|\tilde{\boldsymbol{w}}\|}$ を最大化する問題は，$\|\tilde{\boldsymbol{w}}\|^2$ を最小化する問題に等しい．$\tilde{\boldsymbol{w}}$，$\tilde{b}$ を改めて $\boldsymbol{w}$，b と表記すると (2.8) 式は (2.9) 式に書き換えられ，マージン最大化は (2.9) 式を

解く問題に帰着できる．

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2, \ \text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \ \ (i = 1, ..., n) \tag{2.9}$$

さらに，制約付き最適化問題である (2.9) 式をラグランジュの未定乗数法により，双対問題に置き換える．ラグランジュ乗数 $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)^T$ を用いてラグランジュ関数 L を (2.10) 式と定義する．

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{n} \alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\} \tag{2.10}$$

制約条件が不等式であるため，$\boldsymbol{w}$ の最適解のおいて以下の KKT 条件が適用できる．

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0 \tag{2.11a}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{b}} = 0 \tag{2.11b}$$

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 = 0 \tag{2.11c}$$

$$\alpha_i \geq 0 \tag{2.11d}$$

(2.11c) 式より，$\alpha_i = 0$ または $\alpha_i > 0$ で $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$ が満たされなければならない．ここで，$\alpha_i > 0$ となる $\boldsymbol{x_i}$ をサポートベクトルと呼ぶ．SVM は，教師データ中から，サポートベクトルのみを用いて識別関数を構成する．(2.11a) 式，(2.11b) 式を計算すると，それぞれ (2.12a) 式，(2.12b) 式になる．

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{w_i} \tag{2.12a}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.12b}$$

(2.12a) 式，(2.12b) 式を (2.10) 式に代入すると，$\boldsymbol{\alpha}$ のみで表されるハードマージンの双対問題が得られる．これは線形分離可能な場合のみに適用でき，誤分類を許容しない．

$$\max_{\boldsymbol{a}}\left\{\tilde{L}(\boldsymbol{\alpha})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j}\right\}$$
$$\text{subject to}\sum_{i=1}^{n}\alpha_i y_i=0,\alpha_i\geq 0,\ (i=1,...,n) \tag{2.13}$$

ここで，サポートベクトルに対応する添字の集合を S とすると，(2.12a) 式より，識別関数は (2.14) 式になる．

$$D(\boldsymbol{x})=\sum_{i\in S}a_i y_i \boldsymbol{x_i}^T\boldsymbol{x}+b \tag{2.14}$$

また，(2.11c) 式より，b は (2.15) 式になる．

$$b=y_i-\boldsymbol{x_i}^T\boldsymbol{x}(i\in S) \tag{2.15}$$

(2.14)，(2.15) より得られた識別関数により，$\boldsymbol{x}$ は $D(\boldsymbol{x})>0$ ならクラス K_1 に，$D(\boldsymbol{x})<0$ ならクラス K_2 に分類する．

2.2　ソフトマージン SVM

2.1 節で述べたハードマージン SVM は線形分離可能なことを前提としているが，現実の問題は非線形な問題が多い．そこで非線形な問題に対しても適応できるように，(2.3) 式に非負の変数 $\xi_i(\geq 0)$ を導入する．その式を (2.16) 式に示す．

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geq 1-\xi_i \tag{2.16}$$

これにより，マージンの内側にデータが存在することを許容する．このときソフトマージンの最適化問題は，(2.17) 式，(2.18) 式になる．ここで C は誤分類をどれだけ許容するかを決めるハイパーパラメータであり，小さいほど誤分類を許容し，大きいほど誤分類を許容

しない．よってこの問題は，(2.17) 式の第１項のマージン最大化と第２項の誤分類の許容数のバランスを図る問題である．

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.17}$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \ (i = 1, ..., n) \tag{2.18}$$

2.1 節と同様に，(2.17) 式，(2.18) 式を主問題として，ラグランジュの未定乗数法により双対問題を導出する．ラグランジュ乗数 $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)^T$，$\boldsymbol{\beta} = (\beta_1, ..., \beta_n)^T$ を用いてラグランジュ関数 L を (2.19) 式と定義する．

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{n=1}^{n}\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\} - \sum_{i=1}^{n}\beta_i\xi_i \tag{2.19}$$

制約条件が不等式であるため，$\boldsymbol{w}$ の最適解のおいて以下の KKT 条件が適用できる．

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{w}} = 0 \tag{2.20a}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{b}} = 0 \tag{2.20b}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{\xi}} = 0 \tag{2.20c}$$

$$\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\} = 0 \tag{2.20d}$$

$$\beta_i\xi_i = 0 \tag{2.20e}$$

$$\alpha_i \geq 0, \beta_i \geq 0, \xi_i \geq 0 \tag{2.20f}$$

(2.20a)〜(2.20c) 式を計算すると，それぞれ (2.21a)〜(2.21c) 式になる．

$$\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{w_i} \tag{2.21a}$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{2.21b}$$

$$\alpha_i + \beta_i = C \tag{2.21c}$$

(2.21a)〜(2.21c) 式を (2.19) 式に代入すると，$\boldsymbol{\alpha}$ のみで表されるソフトマージンの双対問題が得られる．

$$\max_{\boldsymbol{a}} \left\{ \tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j} \right\}$$
$$\text{subject to} \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C,\ (i = 1, ..., n) \tag{2.22}$$

識別関数はハードマージン SVM と同じく (2.14),(2.15) 式になる．ただし，(2.20d)，(2.20e),(2.21c) 式により，α_i に関して次の 3 つの場合がある．

2.3　カーネルトリック

カーネルトリック d 次元特徴ベクトルを d'次元特徴空間に写像する

→このときの関数本論文で使うカーネル関数を列挙

データの正規化

一対一法と一対他法について

→一対他法を使う

2.4　多クラス分類への拡張

3　ABC アルゴリズム

3.1　概要

郡知能とは個々での単純な行動が，集団としては高度な振る舞いをすることを模した最適化アルゴリズムである．ABC は 2005 年に karabora によって提案された郡知能の一つである [[3]]．ABC は蜜蜂の採餌行動から着想を得ており，収穫蜂，追従蜂，偵察蜂の 3 種類の蜂によって探索を行う．収穫蜂はすべての食物源に対して探索を行い，追従蜂は評価値の高い食物源を優先して探索を行なう．追従蜂の探索によって評価値の高い食物源の近傍を局所的に探索することができる．収穫蜂と追従蜂による探索で改善されなかった食物源は偵察蜂によってランダムな新しい食物源に置き換えられる．偵察蜂によって局所最適解からの脱出が可能になり，探索範囲を広げることができる．これら 3 種類の蜂による探索を繰り返すことで，より良い解を求めていく．

3.2　探索手順

ABC によって，(3.1) 式のような目的関数を $f(x)$ を最小とする D 次元の x を求める問題を解くことを考える．

$$\min. f(x), \quad \text{subject to } \boldsymbol{x} \in \mathbb{R}^D \tag{3.1}$$

ABC では食物源の数 n と探索上限回数 $limit$ の 2 つのパラメータをあらかじめ設定する必要がある．食物源の数を n とすると，収穫蜂，追従蜂の数はそれぞれ n ずつとなる．$limit$ は偵察バチフェーズで使用する食物源の探索回数の上限である．この値を小さくするとより広範囲の探索に，大きくするとより局所的な探索を行う．各食物源に番号を振り分け，i 番目の食物源を $\boldsymbol{x_i}(i = 1, ..., n)$ とすると $\boldsymbol{x_i}$ の評価値 $fit(\boldsymbol{x_i})$ は (3.2) 式の評価関数により求

められる．

$$fit(\boldsymbol{x_i}) = \begin{cases} \dfrac{1}{1+f(\boldsymbol{x_i})} & \text{if } f(\boldsymbol{x_i}) \geq 0 \\ |1+f(\boldsymbol{x_i})| & otherwise \end{cases} \tag{3.2}$$

step 0 準備

ABC におけるパラメータである食物源の数 n と探索上限回数 $limit$ の設定を行う．次に終了条件の設定を行なう．終了条件としては，アルゴリズムの反復回数が最大反復回数を超えたとき，最良の食物源の評価値もしくは目的関数値が一定値を超えたときなどがある．

step 1 初期化

初期化では，(3.3) 式によって各食物源を各成分の範囲内でランダムに生成する．さらに各食物源の探索回数 $trial_i$ を 0 に初期化する．x_{ij} は $\boldsymbol{x_i}$, の $j(j=1,...,D)$ 番目の成分, $xmin_j$, $xmax_j$ はそれぞれ食物源の各成分の最小値，最大値である．また，$rand$ は一様乱数を表す.

$$x_{ij} = xmin_j + rand[0,1](xmax_j - xmin_j) \tag{3.3}$$

初期化された食物源の中で最も評価値が高いものを $\boldsymbol{x}_{best}$ とし，その食物源のインデックスを i_{best} とする

step 2 収穫蜂フェーズ

収穫蜂フェーズでは収穫蜂が食物源の近傍を探索する．このときの更新式は（3.4）式のようになる．ここで $j,k(k \neq i)$ はランダムに選択される．

$$v_{ij} = x_{ij} + rand[-1,1](x_{ij} - x_{kj}) \tag{3.4}$$

ここで v_{ij} と x_{ij} の評価値を比較し, v_{ij} が x_{ij} よりも優れていたら x_{ij} を v_{ij} に置き換え，$trial_i$ を 0 にリセットする. そうでなければ，$trial_i$ を 1 増やす.

step 3 追従蜂フェーズ

追従蜂フェーズでは収穫蜂フェーズによって更新された食物源の評価値に基づいて，ルーレット選択を行い更新個体を選択する．食物源 x_i の選択確率 p_i は (3.5) 式のようになる．そのため，評価値が高い食物源ほど選択確率が高く，評価値が低い食物源は選択確率が低くなる．食物源の更新は収穫蜂フェーズと同様に行われる．この操作を n 回行う．

$$p_i = \frac{fit(\boldsymbol{x_i})}{\sum_n fit_j} \tag{3.5}$$

step 4 偵察蜂フェーズ

偵察蜂フェーズでは，$trial$ の値が探索打ち切り回数 $limit$ に達した食物源を (3.3) 式によりランダムに生成した新たな解に置換する．置換した食物源の $trial$ を 0 にリセットする．$trial$ の値が $limit$ に達した食物源がなければ何も行わない．

step 5 終了判定

各食物源の評価値 $fit(\boldsymbol{x_i})$ と，$fit(\boldsymbol{x}_{best})$ を比較し，$fit(\boldsymbol{x_i}) > fit(\boldsymbol{x}_{best})$ となる $\boldsymbol{x_i}$ が存在する場合は $\boldsymbol{x}_{best}$ を更新する．その後，終了条件を満たしている場合は最適解をその時の $\boldsymbol{x}_{best}$ として探索を終了し，そうでない場合は **step 2** に戻り，探索を続ける．

4　提案手法

内容 1

5　実験

内容 1

5.1　実験結果

内容 2

6　考察

内容１

7　おわりに

内容 1

謝辞

山口先生有り難う！

参考文献

[1] 近藤 久，浅沼 由馬“人工蜂コロニーアルゴリズムによるランダムフォレストとサポートベクトルマシンのハイパーパラメータ最適化と特徴選択”，人工知能学会論文誌, vol34-2, pp.1-11, 2019.

[2] Cortes, C. and Vapnik, V. Support-vector networks, Ma-chine Learning, Vol.20, No.3, pp.273-297, 1995.

[3] Karaboga, Dervis. An idea based on honey bee swarm for numerical optimization. Vol. 200. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.