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ター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12月25日

目 次

1　はじめに

機械学習モデルには，あらかじめ決めておかなければいけない値であるハイパーパラメータが存在する．これらのハイパーパラメータはモデルの性能に大きな影響を与えるため，適切なハイパーパラメータの選択が必要不可欠である [1]．ハイパーパラメータの例として，分類や回帰に用いられるサポートベクターマシン (SVM) では，ペナルティパラメータ C やカーネル関数の選択やカーネル関数自身がもつパラメータが挙げられる．これらのハイパーパラメータは離散，連続，カテゴリなど様々である．そのため，ハイパーパラメータ最適化（Hyper Parameter Optimization, HPO）では，高次元かつ複雑な探索空間の探索が必要である．さらに，目的関数を評価するためにモデルの学習が必要となるため，多くの場合目的関数の評価が実行時間におけるボトルネックとなる．そのため HPO では評価回数と実行時間がトレードオフの関係にある [2].

従来のハイパーパラメータ調整は手動調整やグリッドサーチ，ランダムサーチで行われてきた．手動でハイパーパラメータを調節することは直感と経験に頼る作業になり，グリッドサーチ，ランダムサーチでは自動化されたものの，探索効率が悪く，高次元の探索空間では計算コストが大きな課題となる [2].

これらの課題を解決するため，より効率的な探索手法として，群知能が注目されている．群知能とは，自然界の生物の群れが高度な振る舞いをすることをコンピュータに適用したアルゴリズムである [3]．群知能の代表的な手法には，人工蜂コロニー（ABC），粒子群最適化（PSO），蟻コロニー最適化（ACO）などがある．特に ABC は，設定パラメータが少なく比較的シンプルなアルゴリズムであるため，HPO の最適化アルゴリズムとして適している．

近藤らの研究 [4] では，カーネル関数を RBF カーネルに固定した上で，ABC を用いて SVM のハイパーパラメータ最適化と特徴選択を行っていた．しかし SVM には様々なカーネル関数が適用でき，それぞれハイパーパラメータが異なる．そのため，カーネル関数の

選択をハイパーパラメータとして扱うことは，より良いSVMモデルの探索を可能にする可能性がある．

そこで本研究では，先行研究で固定されていたカーネル関数を含むハイパーパラメータ全体を探索対象とし，ABCアルゴリズムを用いてSVMのハイパーパラメータ最適化を行う．具体的には，C，4種類のカーネル関数（線形，RBF，シグモイド，多項式），およびそのカーネル関数に対応するパラメータを最適化対象とする．これにより，SVMの分類性能をさらに向上させることを目的とする．

本研究では，KDD'99データセットを使用してSVCクラスのデフォルトパラメータ，先行研究，提案手法との間で比較実験を行った．その結果，先行研究，提案手法ともにデフォルトパラメータよりも良い分類精度が得られた．そして，提案手法の実行時間は先行研究よりも長くなったが，提案手法で得られたパラメータセットは，先行研究で得られたものよりも分類精度が高いという結果が得られた．また，検知率，誤警報率，適合率，F値の値も改善することができ，侵入検知問題に対する性能も向上した．

以降の構成は，第2章でSVMの説明，第3章でABCの説明，第4章で既存手法の説明，第5章で提案手法の説明を行う．第6章では本研究で行った実験の説明と結果を示し，第7章では実験結果からの考察を行う．最後に，第8章では本研究の総括と今後の展望について検討する．

2　サポートベクターマシン (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 [5]．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．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ターを得る．そして，サポートベクター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．

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

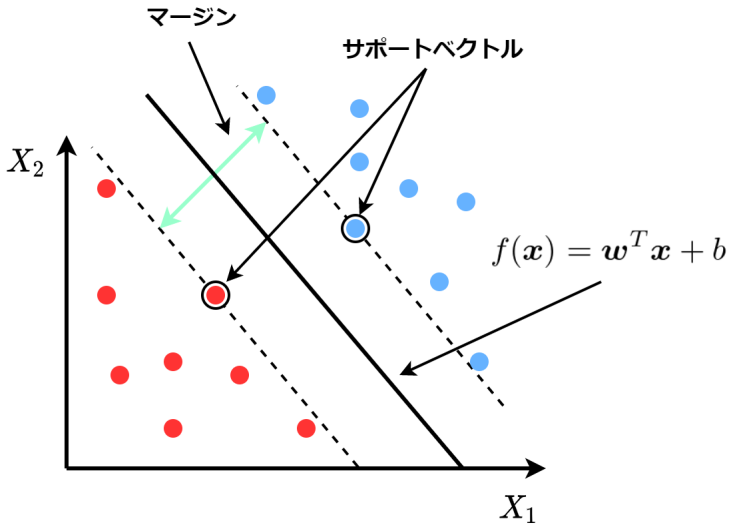

図 1: SVM の分類方法

ここで図 1 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 (サポートベクター) 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，サポートベクター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．

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

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ターである．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 $\boldsymbol{x_i}$ をサポートベクターと呼ぶ．SVM は，教師データ中から，サポー

トベクター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，(2.13a)，(2.13b) 式の，$\boldsymbol{\alpha}$ 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．

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

$$\text{subject to} \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0, \ (i = 1, ..., n) \tag{2.13b}$$

ここで，サポートベクター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，(2.17a) 式，(2.17b) 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，(2.17a) 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．

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

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \ (i = 1, ..., n) \tag{2.17b}$$

2.1 節と同様に，(2.17a) 式，(2.17b) 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 (2.18) 式と定義する．

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{n=1}^{n}\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\} - \sum_{i=1}^{n}\beta_i\xi_i \tag{2.18}$$

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．

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{w}} = 0 \tag{2.19a}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{b}} = 0 \tag{2.19b}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{\xi}} = 0 \tag{2.19c}$$

$$\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\} = 0 \tag{2.19d}$$

$$\beta_i\xi_i = 0 \tag{2.19e}$$

$$\alpha_i \geq 0, \beta_i \geq 0, \xi_i \geq 0 \tag{2.19f}$$

(2.19a)〜(2.19c) 式を計算すると，それぞれ (2.20a)〜(2.20c) 式になる．

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{w_i} \tag{2.20a}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.20b}$$

$$\alpha_i + \beta_i = C \tag{2.20c}$$

(2.20a)〜(2.20c) 式を (2.18) 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 (2.21a)，(2.21b) 式が得られる．

$$\max_{\boldsymbol{a}} \left\{ \tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j} \right\} \tag{2.21a}$$

$$\text{subject to} \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \ (i = 1, ..., n) \tag{2.21b}$$

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．よってソフトマージン SVM とハードマージン SVM の違いは，α_i の上限 C の有無だけの違いである，本研究ではハイパーパラメータである C を最適化対象に含める．

2.3　カーネルトリック

2.2 節で述べたソフトマージン SVM によって，非線形な問題に対しても分類できるようになったが，非線形で複雑な分類問題に対しては高性能な分類器を構成できない．そこで SVM は，N 次元特徴ベクトルを N' 次元特徴空間に写像し，特徴空間上で分離超平面を得る．高次元空間への写像の例を図 2 に示す．この例では，X_1X_2 の軸を新たに追加することによって元の次元では非線形なデータを線形分離可能にしている．一般に，線形分離可能性はデータのサンプル数が多くなればなるほど難しくなり，特徴空間の次元数が大きくなるほど易しくなる．特に，$n-1$ 次元空間では，最大で n 個のデータを任意のラベル付けで線形分離可能である．そのため N 次元特徴ベクトルを N' 次元特徴空間に写像することで

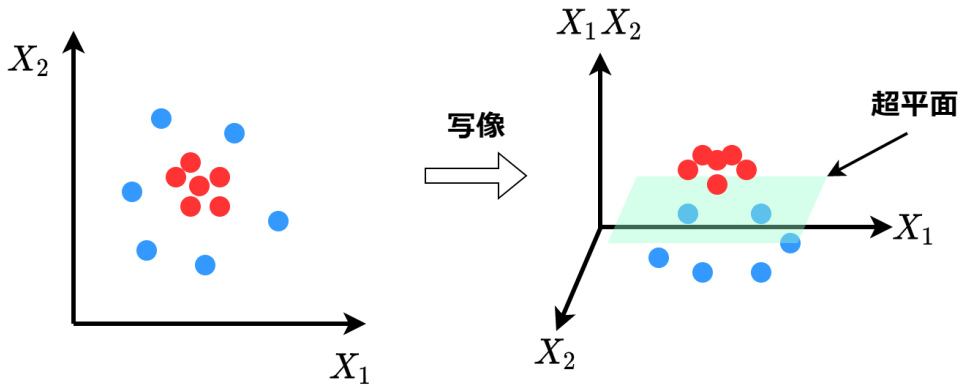

図 2: 高次元空間への写像の例

線形分離可能性が向上する．ここで，N 次元特徴ベクトルを N' 次元特徴空間に写像する関数を $\boldsymbol{\phi}(\boldsymbol{x}) = \{\phi_i(\boldsymbol{x}), ..., \phi_{N'}(\boldsymbol{x})\}^T$ とすると，特徴空間上での分離超平面は (2.22a)，(2.22b) 式の最適化問題を解くことで得られる．

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

$$\text{subject to} \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \ (i = 1, ..., n) \tag{2.22b}$$

しかし次元 N' が大きくなればなるほど $\boldsymbol{\phi}(\boldsymbol{x_i})^T \boldsymbol{\phi}(\boldsymbol{x_j})$ の計算量が膨大になる．そこでカーネル関数を (2.23) 式で定義する．

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{\phi}(\boldsymbol{x_i})^T \boldsymbol{\phi}(\boldsymbol{x_j}) \tag{2.23}$$

もし，(2.23) 式を満たすようなカーネル関数が存在するならば，入力特徴ベクトルの内積計算で $\boldsymbol{\phi}(\boldsymbol{x})$ を得ることができる．すなわち計算量が膨大になる可能性がある $\boldsymbol{\phi}(\boldsymbol{x_i})^T \boldsymbol{\phi}(\boldsymbol{x_j})$ を直接計算する必要がない．このように内積計算をカーネル関数で置き換える手法をカーネルトリックと呼ぶ．本実験では，(2.24)〜(2.27) 式で表されるカーネル関数をハイパーパラメータの対象とする．また，(2.24)〜(2.27) 式内にあるハイパーパラメータは γ，coef0，d

の 3 つである．

$$\text{線形カーネル:}\quad K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i}^T \cdot \boldsymbol{x_j} \tag{2.24}$$

$$\text{RBF カーネル:}\quad K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(-\gamma \|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2\right) \tag{2.25}$$

$$\text{シグモイドカーネル:}\quad K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \tanh(\gamma \boldsymbol{x_i}^T \cdot \boldsymbol{x_j} + \text{coef0}) \tag{2.26}$$

$$\text{多項式カーネル:}\quad K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\gamma \boldsymbol{x_i}^T \cdot \boldsymbol{x_j} + \text{coef0})^d \tag{2.27}$$

2.4　多クラス分類への拡張

SVM は 2 値分類器であるが，1 対他法や 1 対 1 法により多クラス分類への拡張ができる．1 対他法では，各クラスに対し，そのクラスとそれ以外のクラスの分類をする．そのためクラスが K 個ある場合，K 個の SVM モデルを用意する必要がある．一方，1 対 1 法では各クラスのペアの組み合わせに対して分類をする．そのためクラスが K 個ある場合，$\frac{K(K-1)}{2}$ 個の SVM モデルを用意する必要がある．以上より，より計算量が少ないのが 1 対他法，より詳細な分類をするのが 1 対 1 法の特徴である．本研究では一対他法を使用する．

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 [6]．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 $f(\boldsymbol{x})$ を最小とする D 次元の $\boldsymbol{x}$ を求める問題を解くことを考える．

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

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 $\text{fit}(\boldsymbol{x_i})$ は (3.2) 式の評価関数により求

められる．

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

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，$x_{\min_j}$，$x_{\max_j}$ 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．

$$x_{ij} = x_{\min_j} + \mathrm{rand}[0,1](x_{\max_j} - x_{\min_j}) \tag{3.3}$$

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．

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

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．

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

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 $\mathrm{fit}(\boldsymbol{x_i})$ と，$\mathrm{fit}(\boldsymbol{x}_{best})$ を比較し，$\mathrm{fit}(\boldsymbol{x_i}) > \mathrm{fit}(\boldsymbol{x}_{best})$ 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　既存手法

先行研究として，ABC アルゴリズムを用いて SVM のハイパーパラメータ最適化と特徴選択を行った研究がある [4]．先行研究では，カーネル関数を RBF カーネルに固定して SVM のパラメータ C と，RBF カーネルのパラメータ γ の 2 つのパラメータを最適化対象としている．さらに特徴選択を行うため，データセットの特徴数を N とすると個体表現は (4.1) の $N+2$ 次元である．

$$\text{個体表現:}\quad (\text{feature1,feature2,...,feature}N,\ C,\ \gamma) \tag{4.1}$$

そして，各成分の値の範囲はすべて [0,1] であり，feature1,feature2,...,feature$N$ に関しては 0.5 以上でその特徴を使用し 0.5 未満では使用しない．C と γ に関しては，成分の値を a とすると (4.2) 式によって SVM に適用される値に変換される．

$$A = a(a_{max} - a_{min}) + a_{min} \tag{4.2}$$

ここで a_{max}，a_{min} はパラメータの上限値，下限値を表している．この方法により各成分の値の範囲をすべて [0,1] で表すことができるため実装が簡素になる ABC における食物源の評価値 $\mathrm{fit}(\boldsymbol{x_i})$ は (4.3) 式によって計算される．ここで $\mathrm{miss}(\boldsymbol{x_i})$ は SVM が分類を行い，誤分類した割合である．

$$\mathrm{fit}(\boldsymbol{x_i}) = \frac{1}{1 + \mathrm{miss}(\boldsymbol{x_i})} \tag{4.3}$$

5　提案手法

先行研究では，カーネル関数を RBF カーネルに固定してハイパーパラメータ最適化を行っていた．そこで，本研究では SVM のハイパーパラメータ C に加え，(2.24)〜(2.27) 式の 4 つのカーネル関数とそのハイパーパラメータ (γ，coef0，d) を最適化対象とする手法を提案する．最適化アルゴリズムは設定パラメータの少ない ABC を使用する．これにより，ハイパーパラメータの探索範囲を広げ，より良い SVM モデルの探索が可能になる．

5.1　ABC アルゴリズムの適用

(2.24)〜(2.27) 式の 4 つのカーネル関数は，ハイパーパラメータの数や性質が異なっている．そのためカーネル関数が異なる個体では個体表現が変わり，個体の次元数が変化する．しかし，(2.25)〜(2.27) 式の γ や，(2.26)，(2.27) 式の coef0 はそれぞれスケール項，定数項と共通の性質を持っている．そこで，本研究では ABC の個体表現を (5.1) の 5 次元とし，個体表現の変化をその個体の各次元の活性，非活性により表現することとした．ここで，カーネル関数，C は常に使用されるハイパーパラメータで γ，coef0，d はその個体のカーネル関数によって活性，非活性になる．

$$\text{個体表現:}\quad (\text{カーネル関数},\ C,\ \gamma,\ \text{coef0},\ d) \tag{5.1}$$

$C,\ \gamma,\ \text{coef0},\ d$ は [0.1] の範囲の実数とし，(4.2) 式によって SVM に適用される値に変換される．カーネル関数の値は 4 つのカーネル関数を整数値にエンコードしたものである．また，d は離散値であるため，A を四捨五入したものを SVM に適用し，非活性の次元は探索の際に選択されない．表 1 にカーネル関数による活性，非活性の状態の組み合わせを示す．ここで 1 が活性，0 が非活性を表す．

また，ABC による最適化は連続値を前提としているため，カテゴリ変数であるカーネル関数に ABC の更新式は適用できない．そこで本研究では更新次元にカーネル関数が選択さ

表 1: カーネル関数によるパラメータの活性，非活性

カーネル関数	γ	coef0	d
線形カーネル	0	0	0
RBF カーネル	1	0	0
シグモイドカーネル	1	1	0
多項式カーネル	1	1	1

れた際の更新は (5.2) 式の確率 P で更新されるものとする．ここで $\boldsymbol{x_i}$ は更新個体，$\boldsymbol{x_j}$ は $\boldsymbol{x_i}$ と異なるカーネル関数を持つ個体の中からランダムに選ばれた個体である．

$$P = \frac{\mathrm{fit}(\boldsymbol{x_j})}{\mathrm{fit}(\boldsymbol{x_i}) + \mathrm{fit}(\boldsymbol{x_j})} \tag{5.2}$$

ABC における食物源の評価値は (4.3) 式によって計算される．

5.2　提案手法のアルゴリズム

図 3 に提案手法による最適化手順を示す．また，ABC アルゴリズムの詳細な手順を以下に示す．ただし評価値の計算は次の手順で行われる．探索範囲を広げたことにより，学習に時間がかかる個体が生成されるため svm の学習は 100 万イテレーションで打ち切ることとした．評価値は打ち切った時点の評価値を使用する．

1. カーネル関数以外の値を (4.2) 式でマッピング.
2. その食物源が多項式カーネルだった場合 d の値を四捨五入.
3. 学習セットで svm の学習を行う (学習は 100 万イテレーションで打ち切り).
4. 検証セットで svm の分類精度を算出.
5. (4.3) 式で評価値の計算

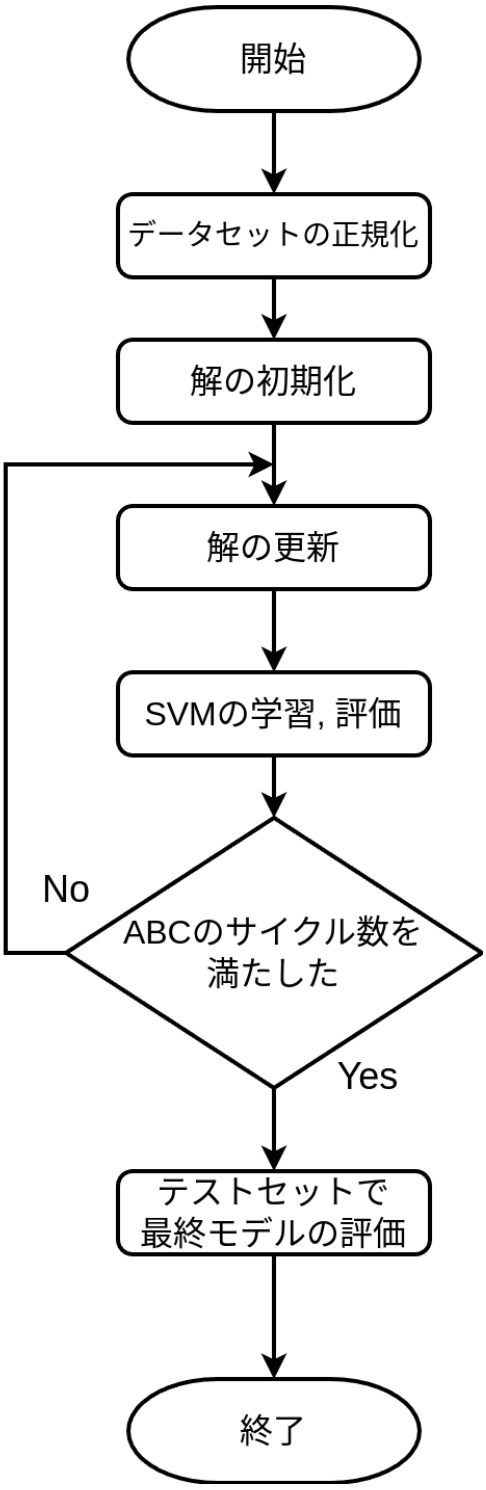

図 3: 提案手法の最適化手順

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 SVN のハイパーパラメータ C，γ，coef0，d の探索範囲の設定を行い，４つのカーネル関数を整数値にエンコードする．最後に終了条件であるアルゴリズムのサイクル数を設定する．

step 1 初期化

初期化では，n 個の食物源をカーネル関数を定める 1 次元目を 1〜4 の整数値でランダムに初期化，その他の 2〜5 次元は [0,1] の一様乱数で初期化する．n 個の食物源の評価値を計算し，初期化された食物源の中で最も評価値が高い食物源とそのインデックスを保存する．

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．

$$v_{ij} = x_{ij} + \mathrm{rand}[-1, 1](x_{ij} - x_{kj}) \tag{5.3}$$

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．

$$p_i = \frac{\mathrm{fit}(\boldsymbol{x_i})}{\sum_n \mathrm{fit}(\boldsymbol{x_j})} \tag{5.4}$$

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 step 1 の初期化と同じ手順でランダムな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

各食物源の評価値と，この時点で最良の食物源の評価値を比較し，$\mathrm{fit}(\boldsymbol{x_i}) > \mathrm{fit}(\boldsymbol{x}_{best})$ 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，指定したサイクル数を満たしていない場合は **step 2** に戻り，探索を続ける．指定したサイクル数に達していた場合は，最適解をその時の $\boldsymbol{x}_{best}$ として探索を終了し，$\boldsymbol{x}_{best}$ を適用した SVM モデルのテストセットに対する分類精度を算出する．

表 2: KDD'99 データセットの内訳

クラス	KDD'99	10%KDD'99
normal	972780	97279
dos	3883370	391458
probe	41102	4107
r2l	1126	1126
u2r	52	52
合計	4898430	494021

表 3: 実験データセットの内訳

クラス	学習	検証	テスト
normal	9740	9650	9766
dos	39127	39238	39106
probe	412	385	410
r2l	120	125	115
u2r	3	4	5
合計	49402	49402	49402

6　実験

侵入検知問題である 10%KDD'99 データセットを使用して，scikit-learn のデフォルトパラメータ，先行研究，提案手法の間で比較実験を行った．先行研究はカーネル関数を RBF カーネルに固定し，ABC アルゴリズムを使用して SVM の C と γ の最適化と特徴選択を行っている [4].

6.1　データセット

10%KDD'99 データセットは，オリジナルの KDD'99 データセットのうちサンプル数の多い normal，dos，probe クラスを 10%に減らしたデータセットである．また，KDD'99 データセットの特徴数は 41 個である．KDD'99 データセットの内訳を表 2 に示す．本研究では，ABC で得られた SVM モデルが未知のデータに対して有効であるかを評価するために 3 つのデータセットを用意した [4]．3 つのデータセットは，SVM の学習に使用する学習セット，SVM の評価に使用する検証セット，最適解を評価するためのテストセットである．ここでテストセットは ABC で得られた最適解を評価するために一度だけ使用される．3 つのデー

表 4: SVC クラスのデフォルト設定

パラメータ	デフォルト値
kernel	RBF
C	1.0
γ	1/n_feature
coef0	0
d	3

タセットを内訳を表 3 に示す．これらのデータセットは 10%KDD'99 データセットからランダムに 10%抽出している．

6.2　実験設定

6.2.1　実験環境

SVM の実装には，CPU:Intel Core i7-12700，メモリ 32GB の計算機上で Python 3.11.0 と scikit-learn 1.5.0 ライブラリの SVC クラスを使用した．SVC クラスにおける本研究で使用するハイパーパラメータのデフォルト設定を表 4 に示す．ここで n_feature はデータセットの特徴数を表す．本研究では n_feature = 38 である．

6.2.2　パラメータ設定

表 5，6 に本研究のパラメータ設定を示す．表 5 の ABC におけるパラメータは先行研究を参照し，両手法ともに同じパラメータ設定である [4]．6 の SVM のパラメータは先行研究では C，γ のみを扱う．提案手法で新たに扱う 4 つの kernel は，SVC クラスにあるすべてのカーネル関数である．coef0，d の探索範囲は試行を重ねた上で調節した．

表 5: ABC の実験パラメータ

コロニーサイズ	20
LIMIT	100
サイクル数	500

表 6: SVM の実験パラメータ

kernel	[linear，RBF，sigmoid，poly]
C	$[10^{-6}$，35000]
γ	$[10^{-6}$，32]
coef0	[0，10]
d	[1，3]

6.3　評価指標

本研究では，得られた SVM モデルの性能評価のため 5 つのクラスに対する分類精度に加えて，検知率，誤警報率，適合率，F 値の 4 つの評価指標を算出する．これら 4 つの指標を得るために必要な値を以下に示す．ここで，通常状態のクラスは normal，攻撃状態のクラスは dos，probe，r2l，u2r である．

- 真陽性 (TP)：攻撃状態として分類され，実際に攻撃状態だったデータ数
- 真陰性 (TN)：通常状態として分類され，実際に通常状態だったデータ数
- 偽陽性 (FP)：攻撃状態として分類され，実際には通常状態だったデータ数
- 偽陰性 (FN)：通常状態として分類され，実際には攻撃状態だったデータ数

TP，TN，FP，FN によって検知率，誤警報率，適合率，F 値はそれぞれ (6.1)〜(6.4) 式で算出される．

$$\text{検知率} = \frac{TP}{TP + FN} \tag{6.1}$$

$$\text{誤警報率} = \frac{FP}{TN + FP} \tag{6.2}$$

$$\text{適合率} = \frac{TP}{TP + FP} \tag{6.3}$$

$$\text{F 値} = \frac{\text{2*検知率*適合率}}{\text{検知率} + \text{適合率}} \tag{6.4}$$

検知率は，攻撃状態を検知できた割合．誤警報率は，通常状態のデータを攻撃状態と分類した割合．適合率は，攻撃状態として分類され実際に攻撃状態だった割合．F 値は検知率と適合率の調和平均である．

6.4　実験結果

以下に示す先行研究と提案手法に関する結果はすべて，それぞれ 10 回ずつ実行した平均値である．各カーネル関数における SVC クラスのデフォルトパラメータ，先行研究，提案手法で比較実験を行った結果を表 7 に示す．ただし，分類精度は小数点第 3 位を，実行時間は小数点第 2 位を四捨五入している．また，線形，RBF，シグモイド，多項式はカーネル関数を指定し，SVC クラスのデフォルトパラメータを使用したものである．加えて，先行研究と提案手法の TP，TN，FP，FN の値を表 8 に，表 8 から算出した検知率，誤警報率，適合率，F 値を表 9 に示す．ただし，表 8 の値は小数点第 2 位を，表 9 の値は小数点第 3 位を四捨五入している．また，表 10〜13 に，先行研究と提案手法で求まった最良解と最悪解を示す．

表 7: 実験結果

	線形	RBF	シグモイド	多項式	先行研究	提案手法
分類精度 [%]	99.68	99.78	96.12	99.76	99.88	99.91
実行時間 [h]	-	-	-	-	11.8	15.5

表 8: 混同行列の値

	先行研究	提案手法
TP	39602.4	39609.7
TN	9743.8	9748.9
FP	22.2	17.1
FN	33.6	26.3

表 9: 性能評価指標の結果

	先行研究	提案手法
検知率 [%]	99.91	99.93
誤警報率 [%]	0.23	0.17
適合率 [%]	99.94	99.96
F 値 [%]	99.93	99.95

表 10: 特徴選択の結果 (最良解: 特徴数 23)

特徴名	選択の有無	特徴名	選択の有無
duration	○	count	○
src_bytes	×	srv_count	○
dst_bytes	○	serror_rate	○
land	×	srv_serror_rate	○
wrong_fragment	○	rerror_rate	×
urgent	○	srv_rerror_rate	○
hot	○	same_srv_rate	×
num_failed_logins	○	diff_srv_rate	○
logged_in	○	srv_diff_host_rate	○
num_compromised	○	dst_host_count	○
root_shell	×	dst_host_srv_count	○
su_attempted	×	dst_host_same_srv_rate	○
num_root	×	dst_host_diff_srv_rate	○
num_file_creations	×	dst_host_same_src_port_rate	×
num_shells	×	dst_host_srv_diff_host_rate	○
num_access_files	×	dst_host_serror_rate	×
num_outbound_cmds	×	dst_host_srv_serror_rate	○
is_host_login	○	dst_host_rerror_rate	×
is_guest_login	×	dst_host_srv_rerror_rate	○

表 11: 特徴選択の結果 (最悪解: 特徴数 21)

特徴名	選択の有無	特徴名	選択の有無
duration	○	count	○
src_bytes	○	srv_count	×
dst_bytes	×	serror_rate	○
land	×	srv_serror_rate	×
wrong_fragment	○	rerror_rate	×
urgent	○	srv_rerror_rate	×
hot	○	same_srv_rate	×
num_failed_logins	○	diff_srv_rate	×
logged_in	○	srv_diff_host_rate	×
num_compromised	×	dst_host_count	×
root_shell	○	dst_host_srv_count	○
su_attempted	○	dst_host_same_srv_rate	○
num_root	×	dst_host_diff_srv_rate	○
num_file_creations	×	dst_host_same_src_port_rate	○
num_shells	×	dst_host_srv_diff_host_rate	×
num_access_files	×	dst_host_serror_rate	○
num_outbound_cmds	×	dst_host_srv_serror_rate	○
is_host_login	○	dst_host_rerror_rate	○
is_guest_login	○	dst_host_srv_rerror_rate	○

表 12: 最良解と最悪解 (先行研究)

	C	γ	分類精度 [%]
最良解	7550.20	0.255	99.897
最悪解	35000.00	15.81	99.848

表 13: 最良解と最悪解 (提案手法)

	C	γ	分類精度 [%]
最良解	469.44	0.485	99.915
最悪解	95.63	0.731	99.907

7　考察

まず，表 7 より，デフォルトパラメータで最も分類精度の高いカーネル関数は RBF の 99.68%であり，先行研究と提案手法の分類精度は 99.88%，99.91%と両手法ともデフォルトパラメータよりも良い分類精度を得ることができた．そして，提案手法は分類精度の面では，先行研究よりも 0.03%良い結果が得られた．一方実行時間の面では提案手法は先行研究よりも約 3.7 時間劣る結果となった．また，表 8，9 より検知率，誤警報率，適合率，F 値のすべてで提案手法のほうが良いという結果になった．そのため分類精度に加えて，侵入検知問題に使用するモデルとしても提案手法のほうが良いパラメータを得ている．

提案手法の実行時間が先行研究の実行時間よりも長くなってしまった原因について考察する．

第一に，提案手法と先行研究の大きな違いとして，特徴選択の有無が挙げられる．特徴選択によって，データセットの次元削減が行われるとカーネル関数の計算が低コストになり，SVM の計算コストが低減する．また，多くの特徴を削減した個体はモデルの精度が低下する可能性があるが，そのような個体は低次元の計算になるため，計算時間は短くなる．ゆえに評価値の低い個体に対して多くの計算資源を使う可能性も低くなる．そのため特徴選択を行ったほうが実行時間は短くなると考えられる．

第二に，提案手法では RBF カーネル以外のカーネル関数も扱い，個体によってカーネル関数が異なる点である．RBF カーネルはハイパーパラメータが 1 個であるが，シグモイドカーネルは 2 個，多項式カーネルは 3 個と RBF カーネルよりも多い，そのためシグモイドカーネル，多項式カーネルはハイパーパラメータ空間が広く，計算コストの高い個体も生成されやすい．また，線形カーネルは元の次元で内積計算を行うため RBF カーネルよりも計算コストは高い．そのため，提案手法の最適化過程では，先行研究よりも評価コストの高い個体を多く評価する必要がある．表 7 のデフォルトパラメータの結果から，RBF カーネルと多項式カーネルの分類精度が高いことが分かるように．提案手法の最適化過程では

RBF カーネルと多項式カーネルの個体がほとんどであった．そのため多項式カーネルの次元数の範囲をさらに拡張して実験を行えば，より実行時間が長くなると考えられる．

次に，提案手法で最終的に得られた最良のモデルがすべて RBF カーネルを使用していたことについて考察する．本研究では先行研究がカーネル関数を RBF カーネルに固定していた点に着目し，4 つのカーネル関数をハイパーパラメータとして扱ったが，最終的には先行研究で固定していた RBF カーネルの個体が最良個体として得られた．そのため，良い個体が生成されやすい RBF カーネルに固定している先行研究の方が良い結果が得られるとも考えられる．しかし，結果として提案手法の方がより良いモデルを生成することに成功している．この理由として，探索の多様性と局所解への陥りやすさが考えられる．先行研究では，RBF カーネルに固定しているため，すべての個体は平均して良い評価値を得る．そして，局所解からの脱出方法は偵察バチフェーズのみである．これに対し提案手法では，カーネル関数が更新される際に，他のカーネル関数で使っていたパラメータをそのまま使用する．そのため，カーネル関数が更新される際には偵察バチフェーズのようなランダム性が生じる．これにより，探索の多様性と局所解からの脱出の面で先行研究よりも優れた手法となった可能性が考えられる．また，提案手法では，個体表現はハイパーパラメータの組の 5 次元だが，先行研究での個体表現はハイパーパラメータに加え，特徴量を含めた 40 次元である．また，提案手法はカーネル関数が変わると探索空間が変化するため，さらに低次元の探索となる．両手法間で探索次元が大きく異なるため，本研究で行った実験では先行研究は十分に探索できていなかった可能性も考えられる．

8　おわりに

本研究では，カーネル関数を含むハイパーパラメータ全体を探索対象とし，ABC アルゴリズムを用いて SVM のハイパーパラメータ最適化を行う手法を提案した．ABC アルゴリズムでカーネル関数を扱うために新たな更新式と探索アルゴリズムを導入した．実験は KDD’99 データセットを学習セット，検証セット，テストセットの 3 つに分けて使用し，SVC クラスのデフォルトパラメータ，先行研究，提案手法の間で比較実験を行った．その結果，提案手法の実行時間は先行研究よりも長くなってしまったが，分類精度の面で先行研究よりも良い結果を得ることができた．また，検知率，誤警報率，適合率，F 値の値も改善することができ，侵入検知問題への性能も向上した．このことから，提案手法は実行時間よりもモデルの性能を重視する環境において有用であると言える．

今後の課題としては， KDD’99 データセット以外のデータセットでも実験を行うことや先行研究よりも長くなってしまった実行時間の短縮を図ることが挙げられる．特にその他のデータセットでも実験を行うことは重要で，提案手法はカーネル関数もハイパーパラメータとして扱ったため，様々なデータセットにおいても良い性能を発揮することが期待できる．

本研究で扱った KDD’99 データセットは比較的大規模なデータセットであるため，SVM の学習を打ち切ったり，多項式カーネルの次元数の探索範囲を低く設定したが，小規模なデータセットにおいてはこれらの制約を設けることなくハイパーパラメータの最適化ができると考えられる．

謝辞

卒業研究および卒業論文の執筆を行うにあたり，終始ご指導を賜りました山口智教授に心より感謝申し上げます．そして，研究活動を行うにあたり有益な助言をくださった院生の先輩方，山口研究室の方々に心より御礼申し上げます

参考文献

[1] M. Feurer and F. Hutter,“Hyperparameter optimization,”Automated Machine Learning, pp.3–33, Springer, 2019.

[2] 尾崎 嘉彦，野村 将寛，大西 正輝,“ 機械学習におけるハイパパラメータ最適化手法：概要と特徴 ”，電子情報通信学会論文誌，vol.J103-D, No.9, pp.615-631, Sep. 2020.

[3] 井上 一哉，鈴木 麻里子,“ 群知能によるパラメータ最適化 ”，土木学会論文集 A2(応用力学), Vol. 74, No. 2 (応用力学論文集 Vol. 21), I_33-I_44, 2018.

[4] 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.

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

[6] 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.