ポケモン対戦において好みのポケモンを活用できるパーティ推薦手法の検討

大室 拓矢[1]　横山 大作[2]

概要：ゲームにおけるチーム構築では、対戦性能に加え、好みのキャラクターを使用したいという要求がある。本研究では、指定されたポケモン 1 体を必ず含む 3 体パーティを推薦する手法を検討する。一部の候補の対戦シミュレーション結果と、構成から算出する 22 特徴量を用いて Strong・Weak の線形 SVM を学習し、Weak モデルで候補を除外した後、Strong モデルの判定スコアで順位付けする。15 種類の固定ポケモンについて、各 250 候補を学習 62 件・テスト 188 件に分け、10 通りの反復ホールドアウトで評価した。平均 Precision@30 は 36.1%で、ランダム基準 27.5%を 8.58 ポイント上回り、平均 Recall@30 は 20.9%となった。上位 30 件に含まれる Strong は平均 10.83 件であり、無作為に選ぶ場合より約 2.57 件多かった。以上より、構築したシミュレーションの評価において、好みの 1 体を採用する制約の下で、有望な組み合わせを効率よく探索するための候補の絞り込みに一定の有効性が示された。

A Study on a Party Recommendation Method for Pokémon Battles Incorporating a User's Preferred Pokémon

Takuya Omuro[1]　Daisaku Yokoyama[2]

Abstract: Players constructing a team may wish to use a preferred character while retaining competitive performance. We study a method for recommending three-Pokémon parties that always include a user-specified Pokémon. Two linear SVMs, Strong and Weak, are trained using 22 composition-based features and labels derived from battle simulations. The method filters candidates using the Weak model and ranks the remaining candidates by the Strong decision score. We evaluated 250 parties for each of 15 specified Pokémon using ten repeated holdout splits, with 62 training and 188 test parties per split. Mean Precision@30 was 36.1%, exceeding the random baseline of 27.5% by 8.58 percentage points, while mean Recall@30 was 20.9%. The top 30 recommendations contained an average of 10.83 Strong parties, approximately 2.57 more than expected under random selection. These results demonstrate the potential of the method to narrow down candidates for efficiently finding promising combinations under the constraint of including a preferred Pokémon, as evaluated within the constructed battle simulator.

1. はじめに

複数のキャラクターを組み合わせて対戦するゲームでは、個々の能力だけでなく、役割の補完や組み合わせを考慮したチーム構築が求められる。一方、プレイヤーの目的は勝利だけとは限らず、外見や設定、愛着などに基づいて好みのキャラクターを使用したいという要求もある。その場合、好みのキャラクターを採用しつつ、その長所を活かし、短所を補う組み合わせを考える必要がある。そこで、好みのキャラクターの採用とチーム全体の対戦性能の両立を目指す構築支援が求められる。

ポケモン対戦でも、タイプ相性、攻撃・防御性能、素早さ、技や特性など複数の要素が関係する。好みのポケモンを 1 体選んだだけでは、残りの個体との補完関係やパーティ全体の性能を容易に判断できない。本研究では、ユーザーが指定した 1 体を必ず含む 3 体パーティを推薦し、候補を絞り込むことを目的とする。ここでは、好みのポケモ

[1] 明治大学大学院
Graduate School, Meiji University
[2] 明治大学
Meiji University

© 2026 Information Processing Society of Japan

ンを採用する制約の下で、残り 2 体との組み合わせによってパーティ全体として有望な構成を探すことに着目する。目指すのは性能評価値の全順位を正確に再現することではなく、高性能と定義した候補を推薦上位に集めることである。

候補数が増えると全候補の対戦シミュレーションには多くの計算を要する。そこで、一部の候補の対戦結果を教師データとし、構成から算出した 22 特徴量を用いる線形 SVM で未学習候補を順位付けする。提案手法では、低性能と推定される候補を Weak モデルで除外した後、Strong モデルの判定スコアを用いて推薦する。

15 種類の固定ポケモンを対象とした評価では、平均 Precision@30 は 36.1%となり、ランダム基準 27.5%を上回った。平均 Recall@30 は 20.9%であり、上位 30 件に含まれる Strong は平均 10.83 件と、無作為に選ぶ場合より約 2.57 件多かった。この結果は、構築したシミュレーション上で、同じ件数の候補を確認する場合に、より多くの有望な組み合わせを発見できることを示している。本稿では、推薦の当たりやすさ、Strong を発見できる範囲、および低性能候補の混入を分析し、好みのポケモンを含むパーティの探索支援としての有効性を検討する。

2. 関連研究

ゲームにおける構築支援では、多数の構成要素から対戦に有効な組み合わせを選ぶことが課題となる。カードゲームのデッキ構築は、そのような組合せ最適化の一例である。

Chen らは、指定された対戦相手に対して有効なデッキを探索する Q-DeckRec を提案した [1]。デッキ構築をカードの入れ替えによる逐次意思決定として定式化し、強化学習によって探索方策を学習する。学習した方策を新たな問題に適用することで、候補のシミュレーション評価を繰り返す探索手法に比べ、推薦時に必要な計算を削減することを目指している。

Kowalski と Miernik は、Legends of Code and Magic のアリーナモードにおけるカード選択方策を、進化的アルゴリズムによって学習する手法を提案した [2]。この問題では、順次提示されるカード候補から選択を行う。各世代のドラフト候補に現れるカードに対応する遺伝子を active gene として扱い、交叉や突然変異の対象を制限することで、他のカードに関する既存の知識を保持しながら学習する。方策の適応度は対戦シミュレーションによって評価する。

これらの研究と本研究は、構成要素の組み合わせを対戦結果に基づいて評価し、学習したモデルや方策を構築支援に利用する点で共通する。本研究では、ユーザーが指定したポケモン 1 体を必ず採用する制約を設け、その条件を満たす 3 体パーティを順位付けする。Q-DeckRec が探索方策を学習し、active gene を用いた手法がドラフトのカード選択方策を学習するのに対し、本研究では完成した候補パーティの 22 の特徴量から Strong・Weak の二値分類器を学習し、判定スコアによって推薦順位を決定する。

また、Q-DeckRec の主たる問題設定は特定の対戦相手に対するデッキ構築であるのに対し、本実験では複数の対戦相手に対する平均評価値を教師ラベルの基準とする。本研究の位置付けは、固定枠の採用制約の下で、静的特徴量による順位付けの有効性を検討することにある。

3. 提案手法

3.1 問題設定と提案手法の概要

ユーザーが指定したポケモン 1 体を固定枠とし、候補集合から固定枠と異なる 2 体を選んで 3 体パーティを生成する。候補集合の個体数を M とすると、固定枠が集合に含まれる場合の組み合わせ数は $\binom{M-1}{2}$ となる。本研究では、プレイヤーが保有する決められた個体の中からパーティを構築する状況を想定する。各個体の技、特性、フォルム、持ち物、能力値などはあらかじめ決められた状態でデータベースに登録されており、推薦ではこれらの設定を変更せず、個体の組み合わせを選択する。

本研究で求めるのは、指定された 1 体を含むという条件を満たし、複数の対戦相手に対する性能評価値が高いパーティである。全候補を対戦シミュレーションで評価すれば比較できるが、個体の組み合わせが増えるほど必要な対戦数も増大する。そこで、一部の個体だけからなる小さなポケモン集合を作り、その中で総当たりのシミュレーションを行って、パーティ構成の特徴量からパーティの勝率の高低を推定できるような軽量な判定器を作成しておく。そして、全ての個体からなるポケモン集合を用いて、指定された 1 体を含むようにパーティ候補を作成し、この判定器を用いて有望なパーティを絞り込むことで、実用的な計算量での推薦を行う、という手法を提案する。

具体的には、パーティ構成から算出する 22 特徴量を入力として、Strong・Weak の二つの線形 SVM の判定スコアを得る。Weak スコアを用いて低性能と推定される候補を除外し、残った候補を Strong スコアの降順に並べた推薦リストを出力する。各モデルは固定枠ごとに学習する。学習済みモデルによる推薦では、候補ごとの追加対戦を必要としない。

3.2 対戦シミュレーションによる性能評価

3.2.1 対戦処理

3 体対 3 体のシミュレータを構築し、実験全体を通して使用する。各パーティはデータベースに登録されている順序で先頭の個体を場に出し、個体が倒れた場合は残存個体を順に繰り出す。技は登録された技から乱数で選択し、状況に応じた処理を行う。行動順は技の優先度、持ち物等の条件、補正後の素早さを用いて決め、行動順に差が無い場合の順序は乱数で与える。ダメージ、状態異常、能力変化、

© 2026 Information Processing Society of Japan

特性、持ち物等は実装した効果に基づいて処理する。最善手を探索する対戦方策ではない。

自発的な交代は、場の 2 体のタイプ相性に基づいて判断し、各チーム最大 2 回までとする。相手の各タイプから自分への倍率から、自分の各タイプから相手への倍率を引いた値を相性スコアとする。現在のスコアが 1 以上で、現在の比較対象より 1 以上小さいスコアを持つ生存個体が控えにある場合に、配列順に候補を更新して交代先を選ぶ。それ以外の判断基準による自発的な交代は実施しないものとする。技による交代フラグが立った場合には別の交代処理を行い、この処理では自発的な交代回数を加算しない。

場にいる 2 体の技による対戦処理は、最大 10 ターンを一単位として実行する。個体が倒れた場合や技による交代が発生した場合、退場または交代を反映する。いずれかのパーティが全滅するか同じ対面が 10 ターン経過するまで処理を繰り返し、時間制限に達した試合は無効とする。技や特性には個別の条件分岐や相互作用があるため、対象個体や効果の拡張には、組み合わせ数の増加に加えて実装・検証の負担が生じる。本稿の評価は、この実装範囲と対戦方策の下で得た結果に限定される。

3.2.2 対戦評価値

パーティ P 側から見た Q との対戦評価値を $s(P,Q)$ とする。対戦終了時の生存個体集合を $L(P)$、$L(Q)$ とし、個体 i の現在 HP を HP_i、最大 HP を $HP_{i,\max}$ として、次式を用いる。

$$s(P,Q) = 100\Bigg(\sum_{i \in L(P)} \frac{HP_i}{HP_{i,\max}} - \sum_{j \in L(Q)} \frac{HP_j}{HP_{j,\max}} \Bigg). \tag{1}$$

控えも含めて各個体の HP 残存割合を合計し、生存個体がいない場合の和は 0 とする。候補 P の性能評価値 $S(P)$ は、有効対戦の相手集合 $V(P)$ に対する平均とする。

$$S(P) = \frac{1}{|V(P)|} \sum_{Q \in V(P)} s(P,Q). \tag{2}$$

タイムアウト等の無効結果は平均から除外する。対戦行列生成時は同一パーティ間のセルの場合を 0 とし、既計算の同一組み合わせはキャッシュから再利用する。逆向きの組み合わせには符号を反転した値を用いる。

3.3 パーティ構成を表す 22 特徴量

対戦シミュレーションを実行せずにパーティの性能を推定するため、各ポケモンの能力値、タイプ、技、特性、持ち物などから 22 の特徴量を算出する。特徴量は、パーティ全体の攻撃・防御・素早さを表す指標に加え、3 体間の能力の偏り、タイプ相性、回復手段、継続ダメージ、能力上昇への対策などを表す指標から構成される。各特徴量の算出内容を表 1 に示す。集中度は、3 体の個体指標の合計に対する最大値の比である。これらの特徴量は実際の対戦における勝率やダメージを直接表すものではなく、パーティ構成の特徴を数値化した近似指標である。各パーティ P の特徴ベクトルを次式で表す。

表 1 学習に使用する 22 の特徴量

列名	算出内容
off_total	追加効果等を含むパーティ攻撃指標
def_total	壁・継戦補正を含むパーティ防御指標
spd_total	補正後の 3 体の素早さ指標の合計を正規化
type	18 タイプからの平均被ダメージ倍率に基づく指標
typedisp	弱点タイプ数と 3 体間の弱点重複に基づく指標
off_max / off_min / off_conc	個体攻撃指標の最大値・最小値・集中度（3 特徴量）
def_max / def_min / def_conc	個体防御指標の最大値・最小値・集中度（3 特徴量）
spd_max / spd_min / spd_conc	個体素早さ指標の最大値・最小値・集中度（3 特徴量）
off_core / off_add	攻撃指標の基礎成分と追加効果成分（2 特徴量）
def_core / def_sustain_mult	防御指標の基礎成分と継戦補正倍率（2 特徴量）
spd_range	個体素早さ指標の最大値と最小値の差
heal_score	回復技・吸収技・回復アイテム等に基づく指標
residual_score	継続ダメージに関係する技・特性・持ち物の指標
stop_setup_score	能力上昇への対策に関係する技・特性の指標

$$\boldsymbol{x}(P) = (x_1(P), x_2(P), \ldots, x_{22}(P)). \tag{3}$$

この特徴ベクトルを Strong・Weak モデルの共通の入力として使用する。学習時には、対戦シミュレーションから得られた性能評価値に基づいて Strong・Weak の教師ラベルを作成し、特徴ベクトルから各ラベルを識別するように学習する。推薦時には、候補の特徴ベクトルを入力し、両モデルの判定スコアを用いて候補の除外と順位付けを行う。

3.4 Strong・Weak モデルの学習

本研究では、高性能なパーティの抽出を目的とする Strong モデルと、低性能なパーティの識別を目的とする Weak モデルの 2 種類の二値分類器を使用する。いずれも 22 の特徴量を入力とする線形サポートベクターマシン（SVM）であり、固定枠ごとに独立して学習する。

まず、シミュレーションによって性能評価値を算出したパーティを学習集合と評価集合に分割する。学習集合に含まれる性能評価値の上位 25%を Strong の閾値、下位 25%を Weak の閾値とする。パーティ P の Strong ラベルおよび Weak ラベルは次式で定義する。

© 2026 Information Processing Society of Japan

$$y_S(P) = \begin{cases} 1 & S(P) \geq T_S, \\ 0 & \text{それ以外}, \end{cases} \tag{4}$$

$$y_W(P) = \begin{cases} 1 & S(P) \leq T_W, \\ 0 & \text{それ以外}. \end{cases} \tag{5}$$

ここで、T_S および T_W は学習集合から求めた Strong・Weak の閾値である。評価集合の正解ラベルにも同じ閾値を適用する。

Strong モデルは Strong ラベルを、Weak モデルは Weak ラベルをそれぞれ教師信号として学習する。学習時には、特徴量に含まれる欠損値や無限大を 0 に置換し、学習データの平均と標準偏差を用いて各特徴量を標準化する。学習した Strong モデルおよび Weak モデルは、入力されたパーティに対して、それぞれ Strong 側、Weak 側への判定の度合いを表すスコアを出力する。本研究では、Strong モデルのスコアをパーティの推薦順位の決定に用い、Weak モデルのスコアを推薦候補の除外に用いる。

3.5 推薦順位の決定

推薦候補に学習時と同じ前処理・標準化を適用する。学習集合に対する Weak モデルの判定スコアの上位 20%を除外閾値とし、この値以上の Weak スコアを持つ候補を除外する。残った候補を Strong モデルの判定スコアの降順に並べ、上位 K 件を推薦する。除外閾値は学習集合から決めるため、テスト候補の除外率は必ずしも 20%にはならない。

Strong モデルが出力するのは Strong クラス側への判定スコアであり、性能評価値そのものの回帰予測や確率ではない。また、Strong でない候補には中間的な性能の候補も含まれるため、非 Strong と Weak は同義ではない。提案手法は、この二つの分類器を一つの推薦処理として使用する。予備的な検討では、Strong スコアだけで順位付けしても低性能候補の混入が残ったため、低性能候補の回避を目的として Weak 除外を組み込んだ。本稿では、この一連の処理を一つの提案手法として評価する。

4. 評価実験

4.1 実験条件

候補選定の参考として使用率情報 [3] を、公開データの取得元として PMSV[4] を用いた。組み合わせの母集団は、使用率情報に基づき上位陣から選定した 50 体とする。各個体は、第 3.1 節で述べた技、特性、フォルム、持ち物、能力値などを設定した状態でデータベースに登録されている。固定枠は、マリルリ、イーユイ、カイリュー、ガブリアス、キョジオーン、サーフゴー、グライオン、ジバコイル、ミミッキュ、サンダー、ドヒドイデ、ディンルー、オーガポン、テツノツツミ、ハッサムの 15 種類とした。

母集団から異なる 3 体を選ぶ全組み合わせから、対戦相手 100 パーティを重複なく無作為抽出する。また、固定枠と異なる 2 体からなる候補全組み合わせから、評価対象 250 パーティを重複なく無作為抽出する。固定枠は順位条件にかかわらず採用し、追加 2 体の候補からは除く。両抽出の seed は 42 とした。評価対象は性能上位 250 件ではない。計 3,750 候補について各 100 対戦相手との結果を用意する。

1 試合の制限時間は 60 秒、1 候補の 100 対戦の処理制限時間は 600 秒とする。時間制限を超えた試合と、候補単位の制限時間到達後の未実行試合を -9999 と記録する。有限な数値であり -9999 でない結果のみを平均し、有効試合が 0 件の場合はエラーとする。有効・無効試合数は記録する。対戦シミュレーションの乱数と、候補抽出・学習分割の seed は区別する。

本実験で用意した対戦枠は、15 種類の固定枠それぞれについて 250 候補と 100 対戦相手の組み合わせからなる計 375,000 枠である。このうち、性能評価値の算出から除外した無効枠は 5,079 枠（1.35%）、有効枠は 369,921 枠（98.6%）であった。ここで、時間制限による無効枠には、1 試合の制限時間超過に加え、候補単位の制限時間に達したため未実行となった対戦も含まれる。

4.2 学習条件とラベル件数

各固定枠の 250 件を学習 62 件・テスト 188 件に無作為分割し、分割 seed を 0〜9 として 10 回評価した。LinearSVC の正則化パラメータ C は、Strong で 3、Weak で 0.03 とした。これらは研究初期の実験において比較的良好な結果が得られた設定を継続して用いたものであり、最適性を確認した値ではない。また、Strong・Weak ラベルを定める上位 25%と下位 25%、および Weak モデルによる候補除外に用いる上位 20%は、探索的に設定した値である。本実験ではこれらの設定を固定し、15 種類の固定枠に共通して適用した。特徴量の欠損値・無限大は 0 に置換するが、性能評価値の欠損・無限大はエラーとする。

Strong・Weak の閾値は学習集合の性能評価値の上位 25%及び下位 25%から求める。標準化も学習集合だけで適合し、テスト集合には同じ変換とラベル閾値を適用する。15 種類× 10 分割のすべてで、学習 Strong・Weak はそれぞれ 16 件であった。テスト Strong は平均 51.75 件、最小 21 件、最大 90 件となった。学習側の四分位点をテストへ適用するため、テスト Strong が必ず 47 件（188 件の 25%）になるわけではない。Weak 除外後の候補数は 126〜176 件であり、すべての条件で上位 100 件まで確保できた。

4.3 評価指標

推薦上位 K 件を R_K、テスト集合内の Strong 集合を G、Weak 集合を W として、次式を用いる。

© 2026 Information Processing Society of Japan

$$\text{Precision@}K = \frac{|R_K \cap G|}{K}, \tag{6}$$

$$\text{Recall@}K = \frac{|R_K \cap G|}{|G|}, \tag{7}$$

$$\text{WeakRate@}K = \frac{|R_K \cap W|}{K}. \tag{8}$$

$K = 10, 20, \ldots, 100$ とする。Precision は提示候補に占める Strong の割合であり、候補確認の当たりやすさを表す。Recall は存在する Strong を拾えた範囲を表し、Weak 混入率は低性能候補が提示される割合を表す。これらはシミュレーション由来のラベルに対する指標であり、実際の対戦勝率や利用満足度ではない。

ランダム基準は、フィルタ適用前のテスト 188 件から一様に K 件を選ぶ場合の期待値とする。Precision の基準は $|G|/188$、Recall は $K/188$、Weak 混入率は $|W|/188$ である。ランダムな推薦リストを多数生成して測定した値ではない。各指標は分割ごとに計算し、各固定枠で 10 回の平均を取り、さらに 15 種類を等重みで平均する。

4.4 補足分析の範囲

特徴量群の除去実験は、先行して評価した 9 種類（第 4.1 節の最初の 9 種類）の補足分析として扱う。各群を除いた後、標準化と両モデルの学習を最初から実施し、ラベル閾値、C、分割などをそろえて比較した。15 種類の主評価とは対象範囲が異なる。

5. 結果と考察

5.1 推薦性能の評価

図 1 に固定ポケモン別の Precision@30 を示す。点は 10 分割平均、横棒は分割間の標本標準偏差であり、信頼区間や独立な実験の標準誤差ではない。全 15 種類の平均 Precision@30 は 36.1%となり、ランダム基準 27.5%を 8.58 ポイント上回った。

15 種類すべてで、固定枠ごとの平均 Precision@30 がランダム基準を上回った。提案手法の値はオーガポンの 27.7%からディンルーの 42.7%まで異なり、固定枠によって改善幅や分割間の変動に違いが見られた。これらの結果から、今回対象とした 15 種類では、好みのポケモンを固定した各条件において、有望な候補を推薦上位に集める傾向が確認できた。

図 2 に示すように、評価したすべての K で平均 Precision はランダム基準を上回った。Precision@10 は 36.7%、Precision@100 は 33.4%であり、少数候補への絞り込みにも一定の有用性がある。ただし、推薦上位のすべてが Strong になるわけではなく、実際の利用では候補の特徴や役割を確認する必要がある。

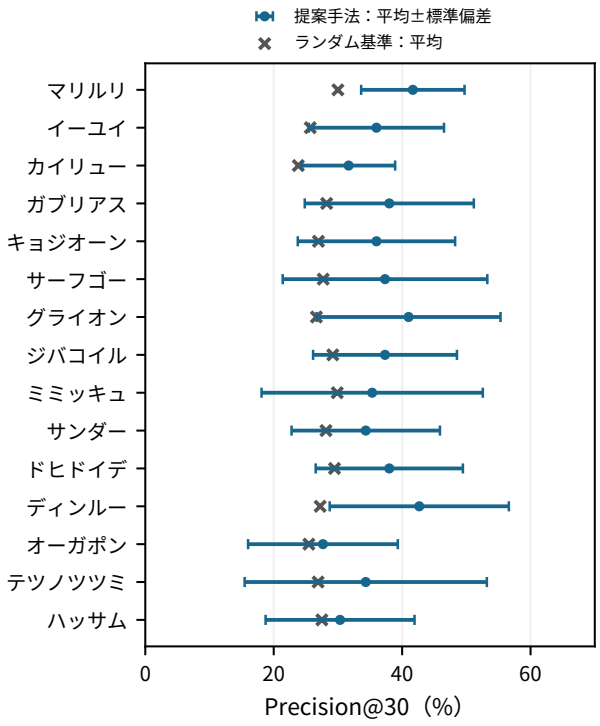


図 1 固定ポケモン別 Precision@30。提案手法は平均と分割間標準偏差、ランダム基準は平均を示す。

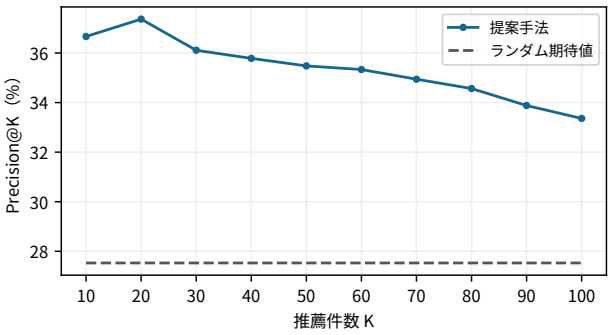


図 2 Precision@K の推移（15 種類・各 10 分割平均）。

5.2 Recall と候補探索の範囲

図 3 に示す平均 Recall@30 は 20.9%であり、ランダム期

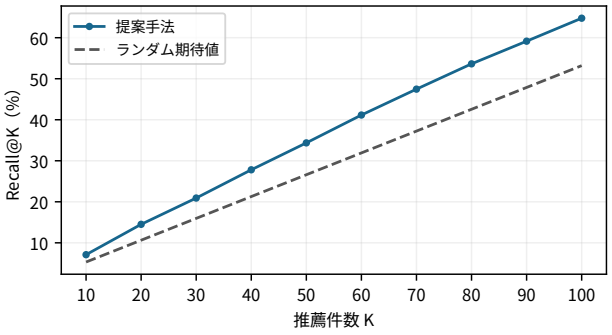


図 3 Recall@K の推移（15 種類・各 10 分割平均）。ランダム期待値は $K/188$。

© 2026 Information Processing Society of Japan

待値 16.0%を上回った。上位 30 件を確認すると平均 10.83 件の Strong が得られ、ランダム基準の平均 8.26 件より約 2.57 件多い。テスト 188 件のうち 30 件、すなわち約 16%の候補を確認することで、Strong の約 20.9%を発見できる結果であった。

推薦件数 K を増やすと、発見できる Strong の範囲も広がった。$K = 100$ では Recall は 64.8%となり、ランダム期待値 53.2%を上回った。したがって、本手法は、少数の候補に絞って有望な構成を探す場合と、確認件数を増やして幅広く候補を探索する場合の双方に利用できる。ただし、$K = 30$ で発見できる Strong は全体の 20.9%であるため、網羅的な探索には推薦件数を増やす必要がある。Precision と Recall を併せて確認することで、候補の当たりやすさと探索範囲の両面から、推薦件数に応じた特性を把握できる。

5.3 低性能候補の混入と推薦漏れ

Weak 混入率@30 は 17.4%であり、無作為推薦の期待値 24.8%より低かった。Strong でない推薦候補には中間群も含まれるため、Precision の補数を Weak 混入率とみなしてはならない。上位 30 件の平均内訳は、Strong10.8 件、Weak5.2 件、中間群 13.9 件である。高性能候補を優先できる一方、低性能候補を完全に回避するものではない。

150 分割を通じた Strong の推薦漏れは延べ 6,138 件であり、そのうち 875 件（14.3%）が Weak モデルで除外され、5,263 件（85.7%）は除外通過後の順位が 30 位を超えていた。この集計には同じパーティの繰り返し出現を含む。後者には推薦枠が 30 件に限られることによる不可避な漏れも含まれ、すべてを順位付けの誤りとはみなせない。

5.4 特徴量に関する補足分析

固定ポケモンを 15 種類に拡張する前に評価した 9 種類を対象として、特徴量群の除去による推薦性能の変化を分析した。この補足分析では、全 22 特徴量を用いた提案手法の Precision@10 は 39.9%、Precision@30 は 37.1%であった。タイプ系 2 特徴量を除くと、それぞれ 34.9%、36.1%となり、上位 10 件では 5.00 ポイント低下した。一方、素早さ系を除くと Precision@30 は 37.6%となった。このように、特徴量群を増やせば一様に推薦性能が上がるわけではなかった。タイプ系を除いた 90 組では、Precision@10 の低下 46 組、同値 23 組、向上 21 組となり、効果は固定枠と分割に依存した。

5.5 具体的な推薦事例

オーガポンを固定した seed = 0 の事例を表 2 に示す。Strong 閾値はこの分割の学習集合から決定し、成功・失敗は推薦上位 30 件を基準とする。表中の順位は Weak 除外後の順位であり、「除外」は推薦順位を持たないことを表す。推薦成功、中間群の推薦、Strong の除外および順位圏外の事例を選び、推薦処理の具体的な挙動を示す。

表 2 オーガポン固定・seed = 0 の事例。固定枠以外の 2 体を記載。

分類	固定枠以外の 2 体	評価値	順位
推薦成功	ドドゲザン グライオン	80.65	23
推薦成功	パオジアン ウーラオス （れんげきのかた）	79.55	29
中間群推薦	イーユイ ドヒドイデ	37.14	16
Weak 除外	アーマーガア グライオン	119.99	除外
Weak 除外	ハッサム アーマーガア	113.93	除外
順位圏外	アーマーガア イダイトウ	109.61	56

中間群推薦の 1 件は中間群、それ以外の 5 件は Strong。

ドドゲザン・グライオンとの組み合わせは Strong であり、23 位として推薦された。一方、イーユイ・ドヒドイデとの組み合わせは中間群であるにもかかわらず 16 位に入った。また、性能評価値 119.99 のアーマーガア・グライオンとの組み合わせは Weak モデルで除外され、109.61 のアーマーガア・イダイトウとの組み合わせは 56 位となった。したがって、モデルの判定スコアは性能評価値の大小そのものを再現せず、高い評価値の候補でも推薦漏れが生じる。

推薦成功例のうち、オーガポン・ドドゲザン・グライオンの構成は、タイプの補完と役割分担の観点からも有望と考えられる。ドドゲザンは、オーガポンの弱点であるひこう技を半減し、どく技を無効にする。また、グライオンはドドゲザンの弱点であるかくとう技を半減し、じめん技を無効にする。攻撃面ではドドゲザンの高い攻撃力と先制技のふいうち、持久戦ではグライオンのどくどく・まもる・みがわりとポイズンヒールを利用できる。このように、相互の弱点を補いながら異なる役割を持つ構成が、Strong として推薦された。ただし、3 体とも物理攻撃に偏るという課題もある。

一方、16 位に推薦されたオーガポン・イーユイ・ドヒドイデの構成は、中間群であった。イーユイの特殊攻撃とドヒドイデの耐久・回復手段によって、物理型のオーガポンを補う役割分担は見られる。しかし、今回設定したオーガポンは持ち物としてかまどのめんを所持しており、ほのおタイプが追加されているため、オーガポンとイーユイはいわ技、イーユイとドヒドイデはじめん技が弱点であり、弱点の重複がある。また、ドヒドイデの特性ひとでなしを活かすには相手を毒状態にする必要があるが、この構成では毒を付与する手段が限られる。したがって、役割の違いという長所があっても、弱点や技・特性の組み合わせまで含めて性能を判断する必要がある。

Strong でありながら Weak モデルによって除外された

© 2026 Information Processing Society of Japan

オーガポン・アーマーガア・グライオンの構成にも、タイプ補完と持久戦の観点から長所がある。アーマーガアはオーガポンのひこう弱点を半減し、どく弱点を無効にする。さらに、グライオンはアーマーガアのでんき弱点を無効にする。アーマーガアのてっぺき・ボディプレス・はねやすめと、グライオンの毒・回復によって、継続的に戦う手段も備える。この構成は性能評価値が 119.99 であったにもかかわらず除外されており、構成上の長所を持ち、シミュレーション上でも高く評価された候補を見逃す場合があることを示している。

以上の事例から、提案手法の推薦成功例には、対戦知識に基づくタイプ補完や役割分担によって、その有望さを説明できる構成が含まれる。一方、弱点の重複などの課題を持つ中間群の推薦や、複数の長所を持つ Strong の除外も生じた。本手法は、好みの 1 体を含む有望な構成を探すための候補提示として利用できるが、推薦結果を選択する際には、各個体の役割や組み合わせも確認する必要があり、より広く組み合わせを検討する場合には、推薦件数を増やして確認することが有効である。

5.6 評価の範囲と今後の展開

本実験は、固定した候補集合と対戦相手集合を対象に、構築したシミュレータの対戦方策の下で推薦性能を評価したものである。Strong・Weak は固定枠ごとの学習分布に基づく相対的な区分であり、固定枠間の絶対性能を直接比較するものではない。また、反復ホールドアウトでは同じ候補が再使用されるため、計 150 回の各評価結果を互いに独立な観測とはみなさない。

第 5.5 節では個体設定に基づいて構成上の長所と短所を考察したが、本シミュレータは技を乱数で選択し、交代もタイプ相性に基づく限定的な方策で行うため、実際のプレイヤーによる対戦とは候補の評価が異なる可能性がある。今後は、より戦略的な対戦方策の下での評価を通じて、推薦構成の妥当性を検証する必要がある。

固定枠別の無効枠数と割合を表 3 に示す。3,750 候補のうち、無効枠を含む候補は 696 件（18.6%）、100 対戦枠すべてが有効な候補は 3,054 件（81.4%）であった。固定枠別の無効率はグライオンの 7.28%が最も高く、他の 14 種類は 0.612〜1.12%であった。グライオンでは全 250 候補に無効枠が含まれており、発生状況は固定枠間で一様ではなかった。

また、候補ごとの有効対戦数は 52〜100 件であり、100 枠中 20 枠以上が無効となった候補は 57 件（1.52%）存在した。全体の無効率は 1.35%であるものの、一部の候補では対戦相手の約半数が平均の計算から除かれている。このため、全体率だけから性能評価値への影響が小さいとは判断できない。処理時間の長い相手や対戦展開に無効枠が偏る場合、有効試合の平均が、予定した 100 対戦相手すべてに対する平均とは異なる可能性がある。これにより Strong・Weak の閾値やラベル、推薦性能の評価に影響する可能性があるが、除外された対戦の結果が不明なため、その方向と大きさは現段階では特定できない。

表 3 固定枠別の無効対戦枠。各固定枠の総数は 25,000 枠。

固定ポケモン	無効枠数	無効率（%）
マリルリ	268	1.07
イーユイ	178	0.712
カイリュー	185	0.740
ガブリアス	219	0.876
キョジオーン	217	0.868
サーフゴー	262	1.05
グライオン	1,819	7.28
ジバコイル	153	0.612
ミミッキュ	269	1.08
サンダー	276	1.10
ドヒドイデ	268	1.07
ディンルー	189	0.756
オーガポン	241	0.964
テツノツツミ	280	1.12
ハッサム	255	1.02
全体	5,079	1.35

この評価範囲において、15 種類すべての固定枠で平均 Precision@30 がランダム基準を上回り、好みのポケモンを含む候補から有望な組み合わせを優先して提示できた。少数の推薦候補を確認することで Strong を効率よく発見でき、推薦件数を増やすことで探索範囲を広げられる点が、本手法の利点である。

今後は、候補集合や対戦相手集合を変更した評価を行い、適用範囲を広げる。また、シミュレータの継続的な検証と対戦方策の改善、無効対戦の集中要因と、それに伴う評価値・推薦性能への影響の検証、特徴量および学習条件の検討を進め、推薦性能の向上を図る。さらに、推薦候補の特徴や選定理由を提示することで、好みのポケモンを含むパーティの比較と選択を支援する。

6. おわりに

本研究では、好みのポケモン 1 体を採用する制約の下で、パーティ全体として有望な組み合わせを探すため、22 特徴量と Strong・Weak の線形 SVM を用いた推薦手法を検討した。15 種類の評価では、平均 Precision@30 は 36.1%、平均 Recall@30 は 20.9%となった。上位 30 件には Strong が平均 10.8 件含まれ、ランダム基準より約 2.6 件多く、同じ確認件数で有望な候補をより多く発見できた。以上より、構築したシミュレーション上で、好みのポケモンを含む構成の絞り込みに一定の有効性が示された。

今後は、シミュレータの検証と対戦方策の改善、異なる

© 2026 Information Processing Society of Japan

候補集合・対戦相手集合での評価、特徴量と学習条件の検討を進める。さらに、推薦理由の提示や候補の多様性を考慮し、好みのポケモンを採用しながら有望な構成を探せるパーティ構築支援へと発展させる。

参考文献

[1] Z. Chen, C. Amato, T.-H. D. Nguyen, S. Cooper, Y. Sun, and M. Seif El-Nasr, "Q-DeckRec: A Fast Deck Recommendation System for Collectible Card Games," 2018 IEEE Conference on Computational Intelligence and Games (CIG), 2018.

[2] J. Kowalski and R. Miernik, "Evolutionary Approach to Collectible Card Game Arena Deckbuilding using Active Genes," 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, 2020, doi: 10.1109/CEC48606.2020.9185755.

[3] ポケモンバトルデータベース,「ポケモン使用率ランキング（シーズン 13・シングル）」, https://sv.pokedb.tokyo/pokemon/list?season=13&rule=0,（2026 年 7 月閲覧）.

[4] Ruimusume, "PMSV," GitHub, https://github.com/Ruimusume/PMSV,（2026 年 7 月閲覧）.

© 2026 Information Processing Society of Japan