

# データ分析プロジェクト 解説資料①

## 対戦型RPGのタイプ相性ネットワーク分析

名古屋大学大学院 情報学研究科 複雑系科学専攻  
多自由度システム情報論講座

修士1年 鈴木琳久

# はじめに

- 本資料について

対戦型RPGのタイプ相性について、ネットワーク分析の観点から考察するプロジェクトの概要、結果、解説をまとめた資料になります。

- 制作物について

本プロジェクトで作成したコード、ファイル、グラフは、GitHub上で一般公開しています。

[https://github.com/rikuli-35/seminar-materials/blob/main/02\\_data\\_analysis/type\\_analysis.md](https://github.com/rikuli-35/seminar-materials/blob/main/02_data_analysis/type_analysis.md)

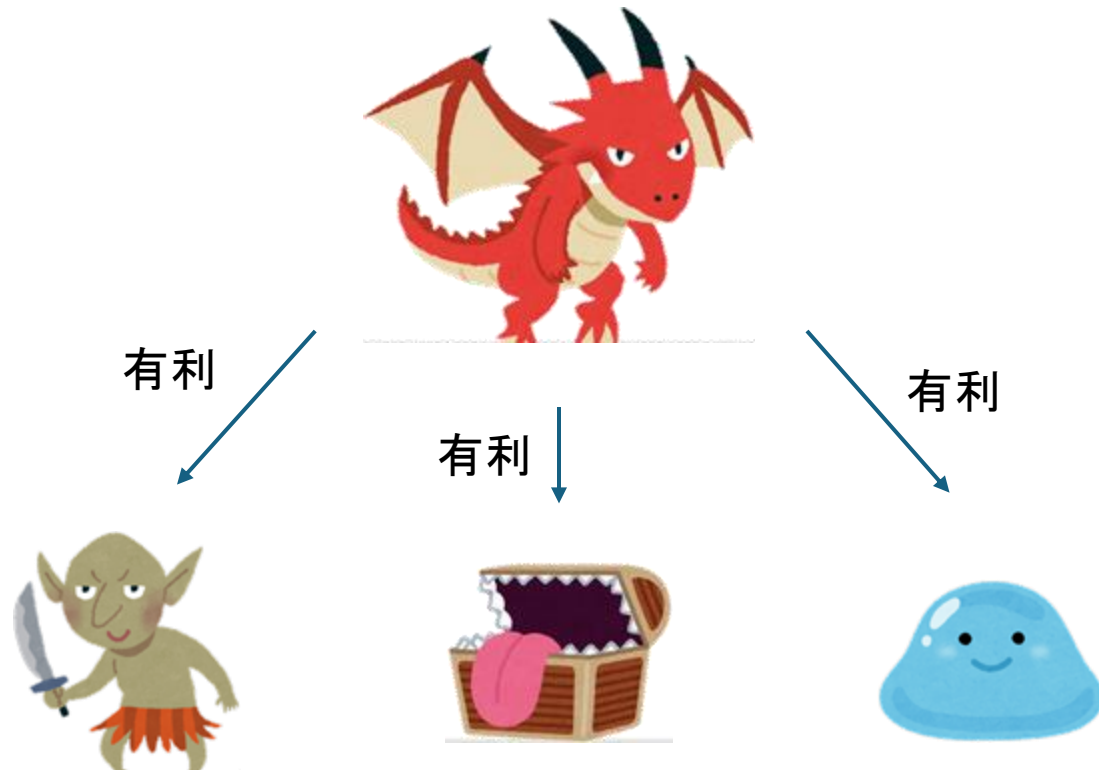
# 本プロジェクトの流れ

- 本プロジェクトの動機、意義、目標について
- ネットワーク分析における必要な知識の紹介
- 本プロジェクトでのルール
- チュートリアルを紹介、結果、評価
- (新タイプの提案、再実験、結果、評価)
- 参考文献、既存の研究の紹介

# 本プロジェクトの動機

- 「Kaggle」というデータ分析プラットフォームの存在を知り、ゲームに関するデータセットに興味を持った
- 大学院の研究でネットワークに関連した内容を扱うため、その前段階として実践経験を積みたかった
- ゲーム制作において、データ分析がどのように活かせるのか、また、どのようなことを意識しなければならないのか、自分なりに考えてみたかった
- 「面白さ」という個人によって異なる感覚を論理的に説明し、作り出すことに興味があった

# 対戦型ゲームにおけるゲームバランスの重要性



特定のタイプのみが強すぎると、、、



対戦が面白くない！



好きなキャラが弱くて使えない！

# 対戦ゲームの「公平性」と「多様性」のバランス

このように、「(タイプ相性の) **公平性**」が保たれていない対戦は面白くない！

では、ゲームの公平性が完璧に保たれたゲームは面白いのか？

→「このタイミングで強いタイプを出す」といった**戦略**や**読み合い**が無くなってしまう、、、

→「(タイプ相性の) **多様性**」も保たれていないといけない！

つまり、対戦環境では、「公平性」と「多様性」の**バランスが重要**！

# 本プロジェクトの意義

- 新しい内容が追加されたり、対戦のルールが変更されたとき、ユーザーのゲーム体験が大きく変化する
- 現状のゲームバランスが良くても、新しい内容を追加しないと、ユーザーが飽きてしまう可能性がある
- そのため、ゲームバランスを保ちながら、新しい体験をユーザーに提供し続けなければならない

# 本プロジェクトの目標

- 「ポケットモンスター」を題材として、対戦型RPGのゲームバランスがどのように構成されているかを考察する ※補足①
- ポケモンの「世代」ごとにタイプ相性が異なるため、世代間ごとにタイプ相性ネットワークを構築する
- 世代の変化とともにゲームバランスがどのように改善されているのかをスコア化する
- (現状のゲームバランスを保ったまま、新しいタイプを提案し、再度ネットワーク分析を行って変化を考察する)

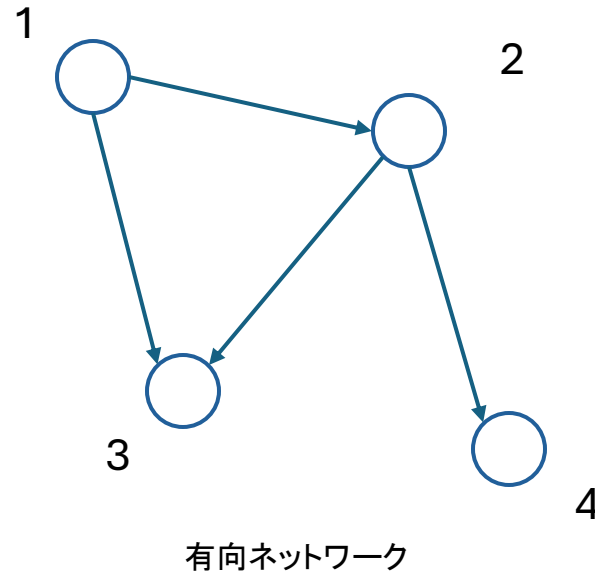
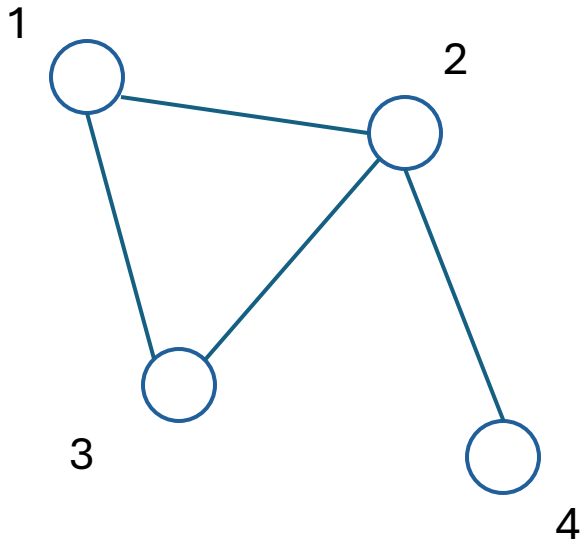


# ネットワーク分析の基礎知識①

- ネットワークとは？

頂点(ノード)とそれらをつなぐ枝(エッジ)から構成される構造のこと。

枝に向きがついている場合は、**有向**ネットワークと呼ばれる。



$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

無向ネットワーク(隣接行列)

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

有向ネットワーク(隣接行列)

# 本プロジェクトで構築するネットワーク

(例) ポケモンのタイプ相性ネットワークの場合

- 各ノードは「ポケモンのタイプ」を表す
- 「攻撃技のタイプ → 攻撃を受けるポケモンのタイプ」のように、タイプの組み合わせごとに方向性があるため、枝は「向き」をもつ（有向ネットワーク）
- タイプの組み合わせによってダメージ倍率が異なるので、枝に「**重み**」(枝の重要性の指標)をつける

このように構築されるネットワークは、「**重み付き有向ネットワーク**」となる

# ネットワーク分析の基礎知識②

- 次数(次数中心性)とは？

頂点をもつ枝の数のこと

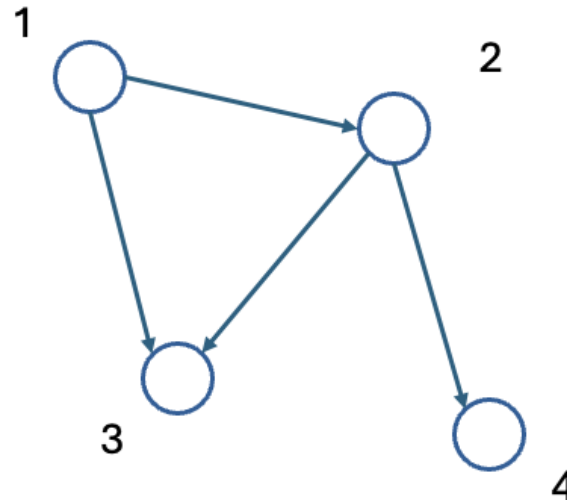
(有向ネットワークの場合は、入次数と出次数に区別する)

- 入次数

頂点に入ってくる枝の数

- 出次数

頂点から出ていく枝の数



頂点2について  
入次数は1  
出次数は2

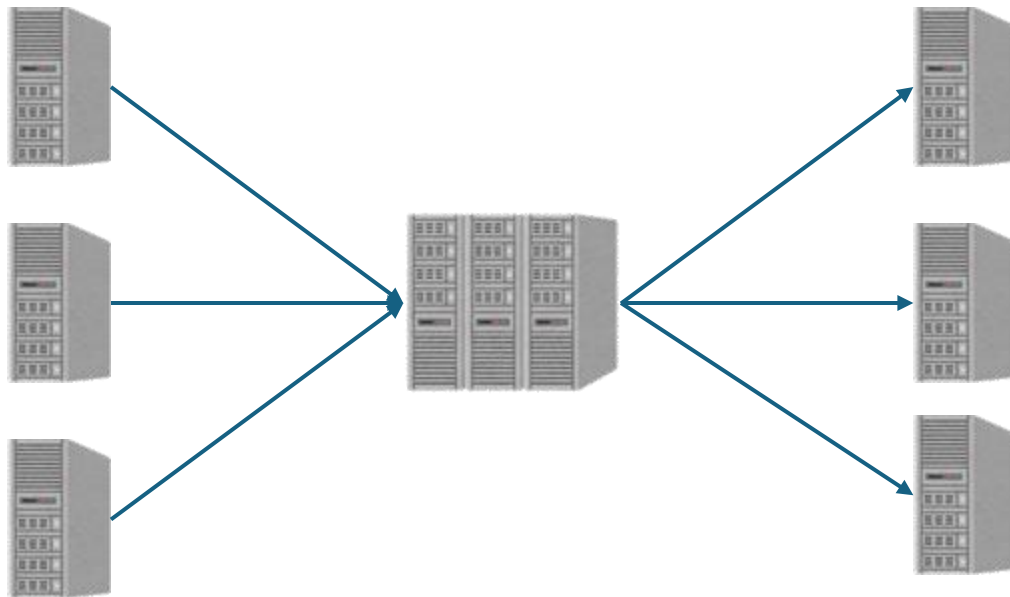


# ネットワーク分析の基礎知識③

- 媒介中心性とは？

その頂点がネットワーク上の流れにどの程度関わるのかを表す指標。

情報の中継点となるノードは媒介中心性が高くなる。



大きいサーバはネットワーク上で  
重要な役割がある！



# ネットワーク分析の基礎知識④

- 固有ベクトル中心性とは？

あるノードの重要性は、それとつながっているノードの重要性の高さに基づいて決まる。



知り合い



この人も有名人である  
可能性が高い！

# 本プロジェクトでのルール

本プロジェクトでは、以下の6つのルールを使用します ※補足②

入次数が高い → 「弱点が多い(弱い)タイプ」

出次数が高い → 「相手に弱点をつきやすい(強い)タイプ」

- ルール①

次数中心性は、そのタイプの単純な「強さ」の指標であると解釈する。

- ルール②

次数中心性のスコア(※補足③)が高いほど、「公平性」が高いと評価する。

# 本プロジェクトでのルール

媒介中心性が高い → 「ネットワーク上の中継地点になりやすいタイプ」

固有ベクトル中心性が高い → 「強いタイプとつながっているタイプも強くなる」

- ルール③

媒介中心性が高いほど、(パーティ編成等で)採用されやすいタイプと解釈する。 ※補足④

- ルール④

固有ベクトル中心性が高いほど、メタになりやすいタイプと解釈する。 ※補足⑤

- ルール⑤

媒介・固有ベクトル中心性のスコア(※補足⑥)が高いほど、「多様性」が高いと評価する。

# 本プロジェクトでのルール

- ルール⑥

次数中心性のスコアと媒介・固有ベクトル中心性のスコアを統合した総合スコアをその世代のゲームバランスのスコアとする。

総合スコアは「公平性」と「多様性」の重要性が1:1になるようにスコア化する。

具体的には、以下のように算出するものとする。 ※補足⑦

「ゲームバランスのスコア」＝

$$\text{「次数中心性のスコア」} \times 0.5 + \text{「媒介・固有ベクトル中心性のスコア」} \times 0.5$$



# 実行環境・コード解説((R言語))

以上のルールを評価指標として、ネットワーク分析を行います。

初代タイプ相性を例として、ネットワークの構築やグラフの作成をR言語で行いました。

コード解説(チュートリアル)はこちら:[https://rikuli-35.github.io/seminar-materials/02\\_data\\_analysis/type\\_analysis\\_tutorial.html](https://rikuli-35.github.io/seminar-materials/02_data_analysis/type_analysis_tutorial.html)

実行環境: RStudio Desktop macOS 13+ または Windows 10/11

ダウンロードはこちら:<https://posit.co/download/rstudio-desktop/>

# 使用するデータセット

- 初代のタイプ相性表

<https://wiki.ポケモン.com/wiki/相性>

- 第2世代～第5世代のタイプ相性表

[https://www.nintendo.co.jp/ds/ipkj/basis/pdf/pokemonHGSS\\_typeTable.pdf](https://www.nintendo.co.jp/ds/ipkj/basis/pdf/pokemonHGSS_typeTable.pdf)

- 第6世代のタイプ相性表

<https://www.pokemon.co.jp/ex/xy/battle/01.html>

- 上記をもとに自作したcsvファイル

[https://github.com/rikuli-35/seminar-materials/blob/main/02\\_data\\_analysis/type\\_1.csv](https://github.com/rikuli-35/seminar-materials/blob/main/02_data_analysis/type_1.csv) 補足※⑧

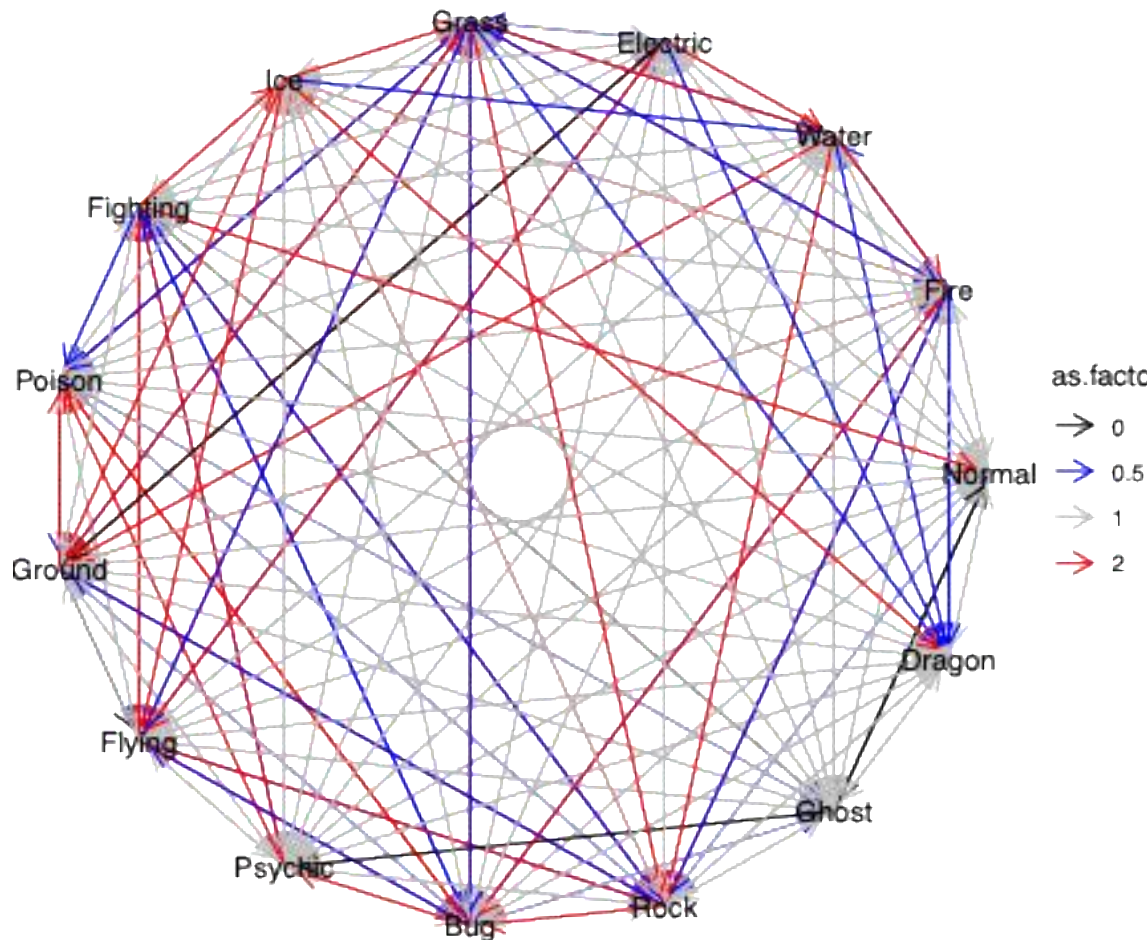
チュートリアルで使します



[https://github.com/rikuli-35/seminar-materials/blob/main/02\\_data\\_analysis/type\\_2\\_to\\_5.csv](https://github.com/rikuli-35/seminar-materials/blob/main/02_data_analysis/type_2_to_5.csv)

[https://github.com/rikuli-35/seminar-materials/blob/main/02\\_data\\_analysis/type\\_6\\_to\\_9.csv](https://github.com/rikuli-35/seminar-materials/blob/main/02_data_analysis/type_6_to_9.csv)

# チュートリアルの実行結果①



構築した有向ネットワーク

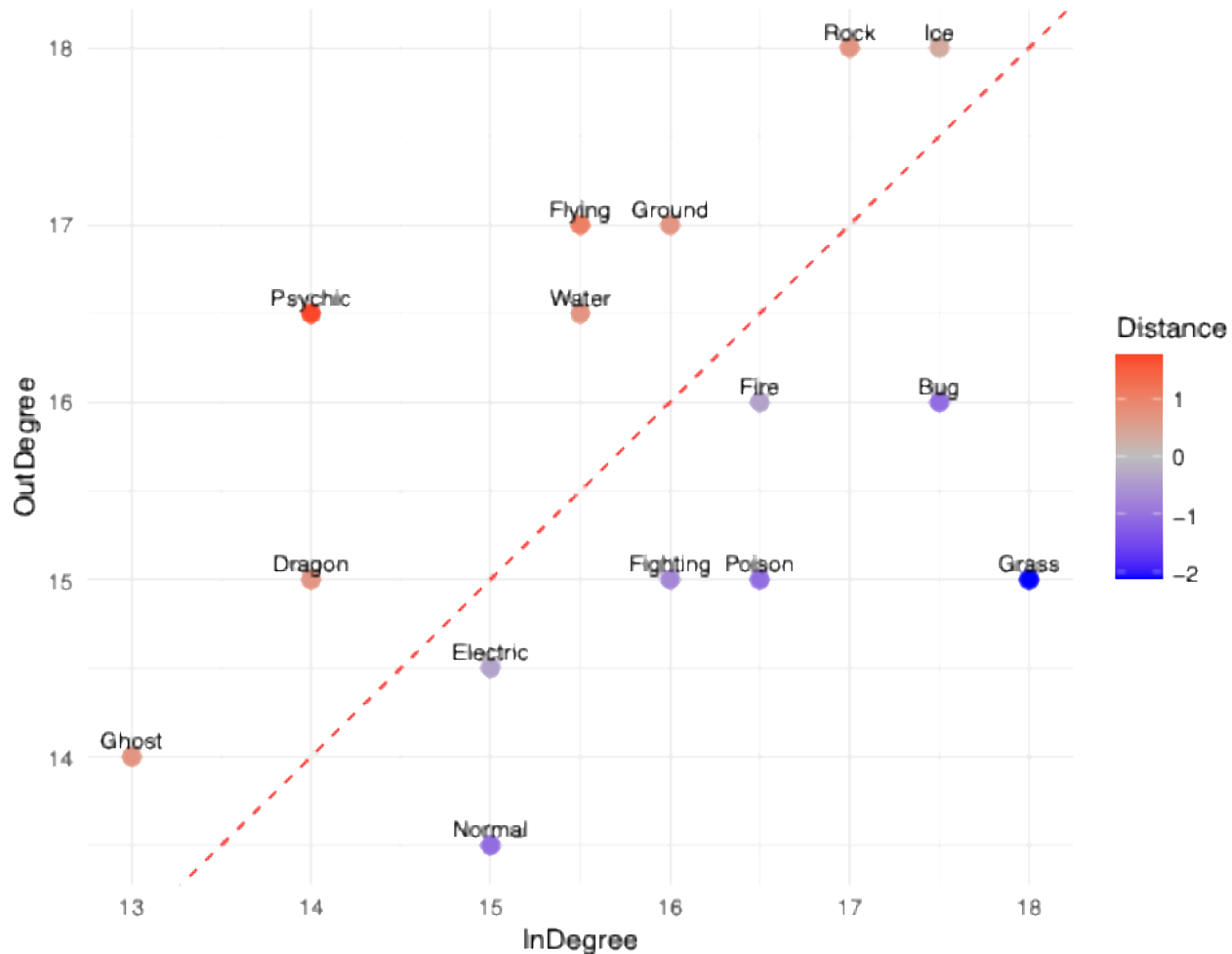
##	Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison
##	15.0	16.5	15.5	15.0	18.0	17.5	16.0	16.5
##	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	
##	16.0	15.5	14.0	17.5	17.0	13.0	14.0	

タイプ別 入次数

##	Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison
##	13.5	16.0	16.5	14.5	15.0	18.0	15.0	15.0
##	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	
##	17.0	17.0	16.5	16.0	18.0	14.0	15.0	

タイプ別 出次数

## チュートリアルの実行結果②



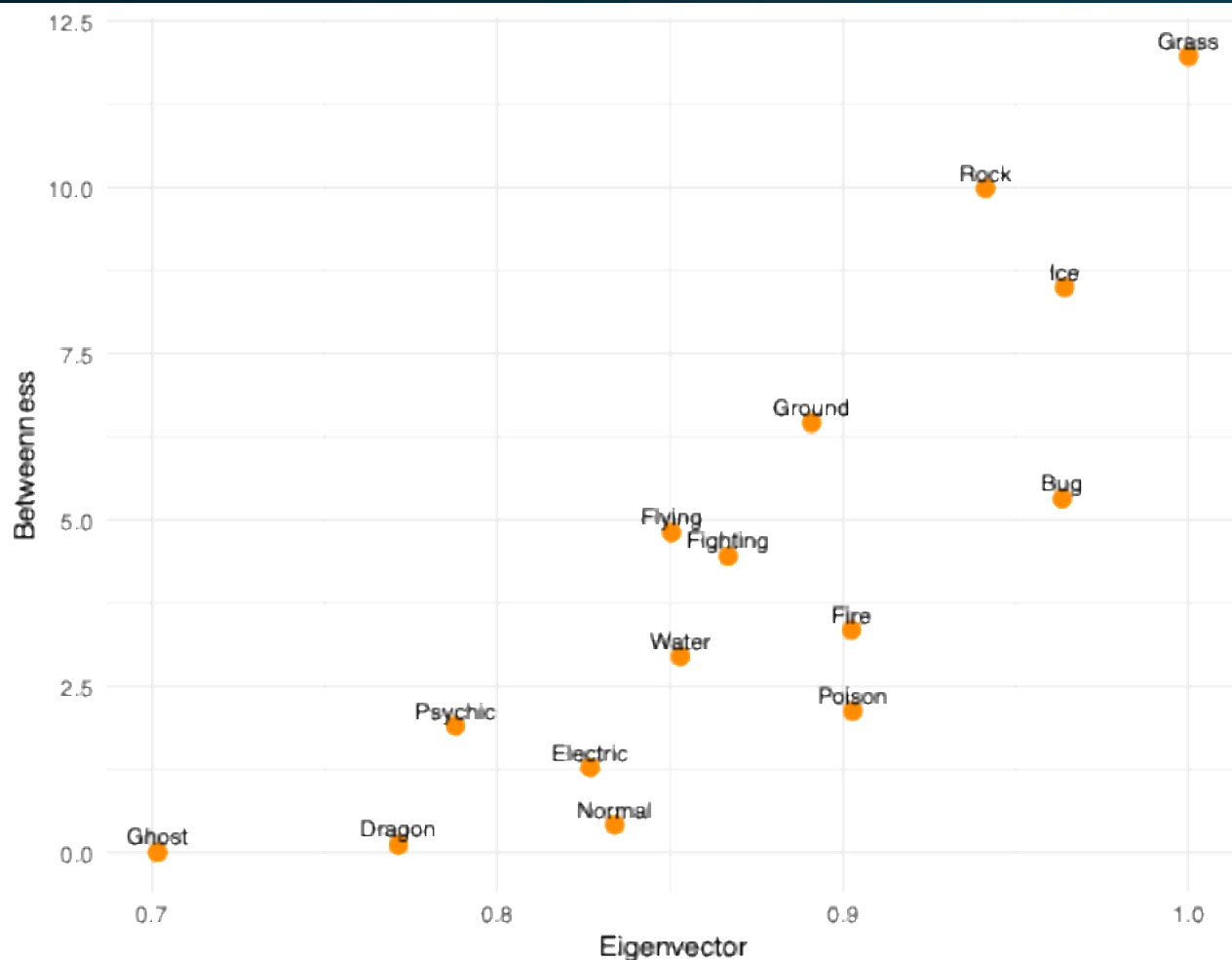
次数中心性の散布図

- 縦軸は入次数、横軸は出次数を表す
- 赤色の点線は、入次数 = 出次数を表す直線
- 各ノードの色が点線との距離を表す  
(この距離は符号付き距離)

点線より上側にある点は  
出次数 > 入次数  
だから強いタイプと言える！



# チュートリアルの実行結果③



- 縦軸は媒介中心性、横軸は固有ベクトル中心性を表す

##	Grass	Rock	Ice	Ground	Bug	Flying	Fighting
##	11.9694444	9.9833333	8.4944444	6.4611111	5.3166667	4.8111111	4.4500000
##	Fire	Water	Poison	Psychic	Electric	Normal	Dragon
##	3.3444444	2.9444444	2.1250000	1.9027778	1.2777778	0.4166667	0.1111111
##	Ghost						
##	0.0000000						

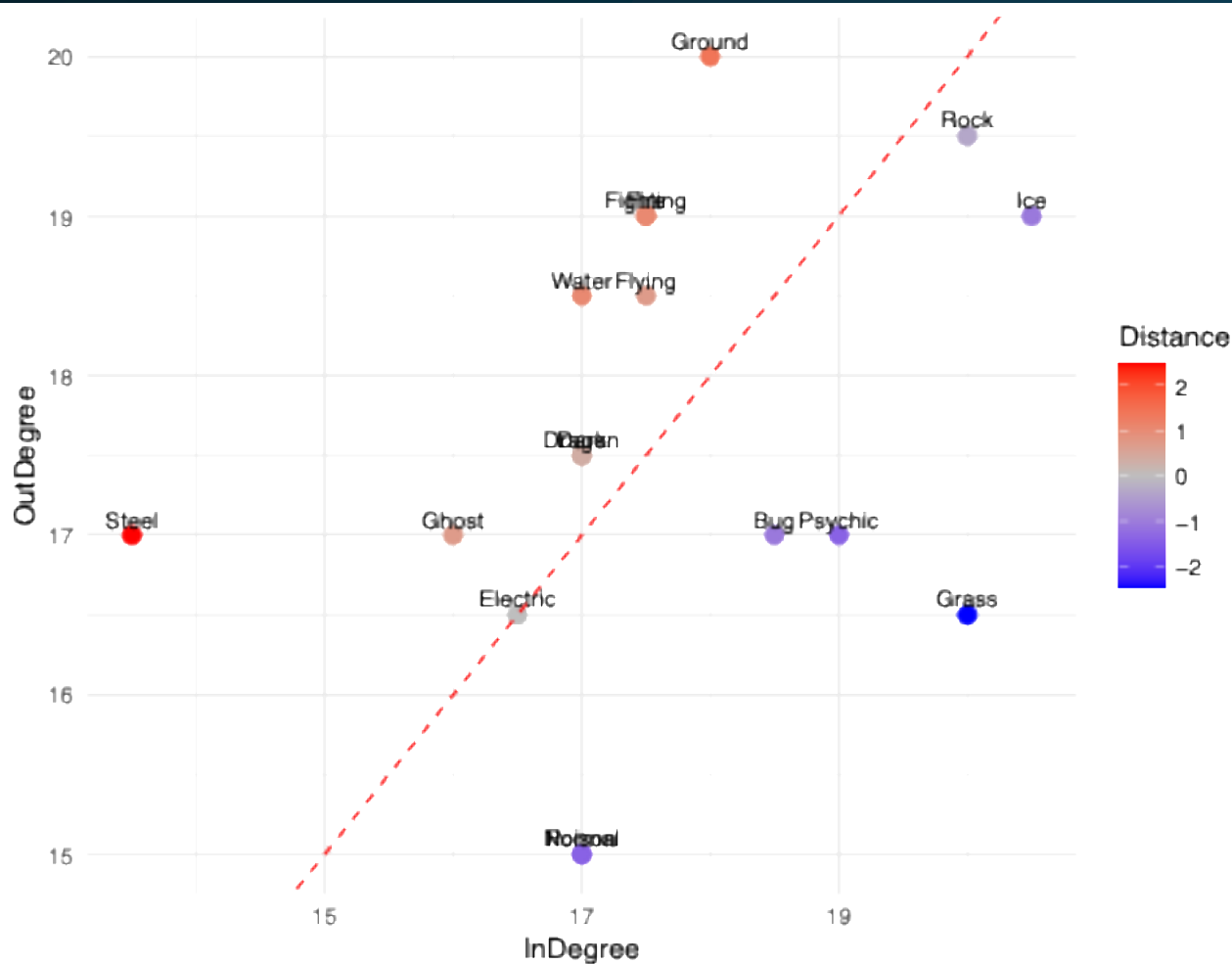
タイプ別 媒介中心性

##	Grass	Ice	Bug	Rock	Poison	Fire	Ground	Fighting
##	1.0000000	0.9640843	0.9634309	0.9412123	0.9028351	0.9025043	0.8909043	0.8667929
##	Water	Flying	Normal	Electric	Psychic	Dragon	Ghost	
##	0.8529862	0.8504045	0.8339442	0.8268267	0.7878577	0.7713628	0.7016588	

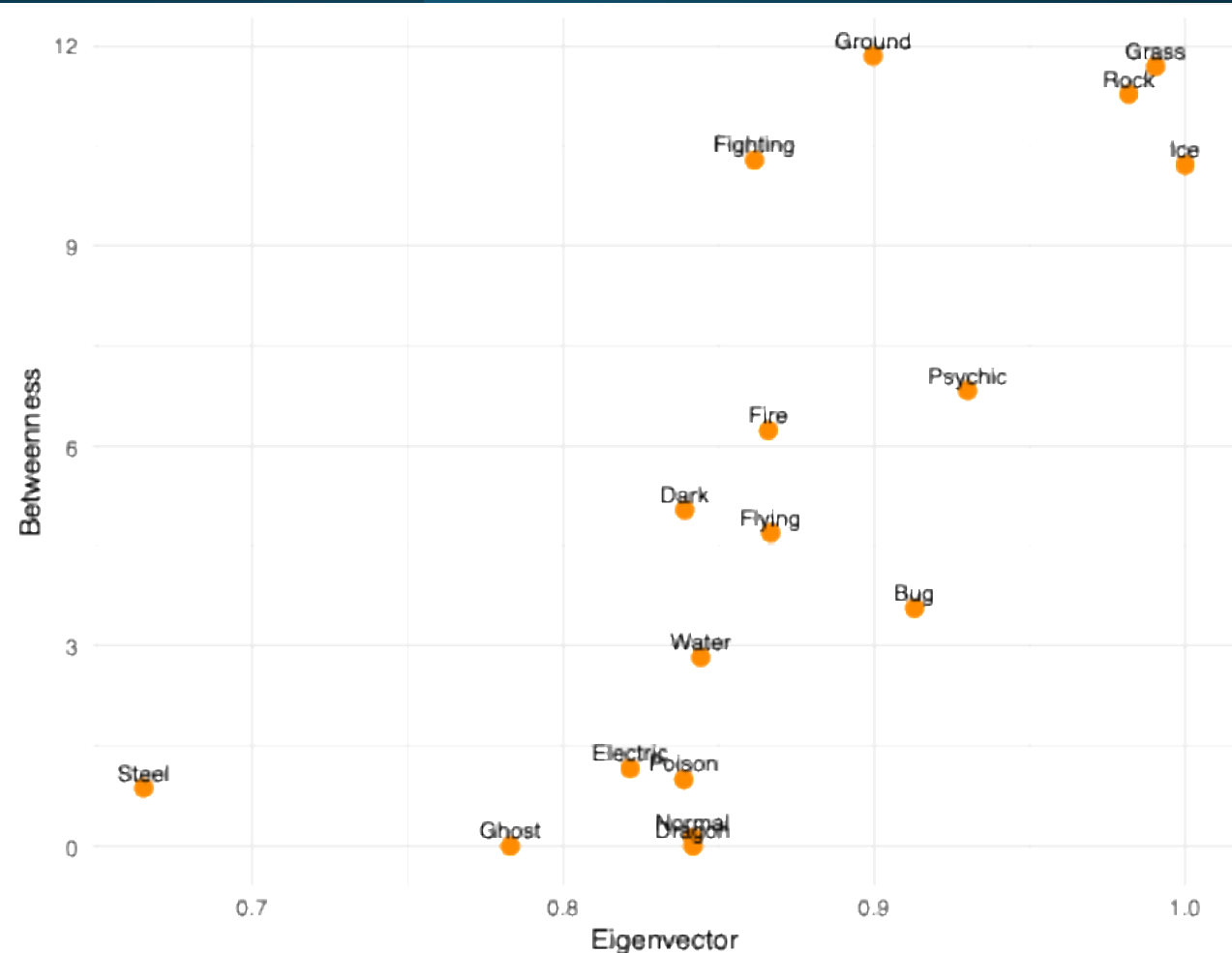
タイプ別 固有ベクトル中心性

媒介・固有ベクトル中心性の散布図

# 第2世代～第5世代での結果

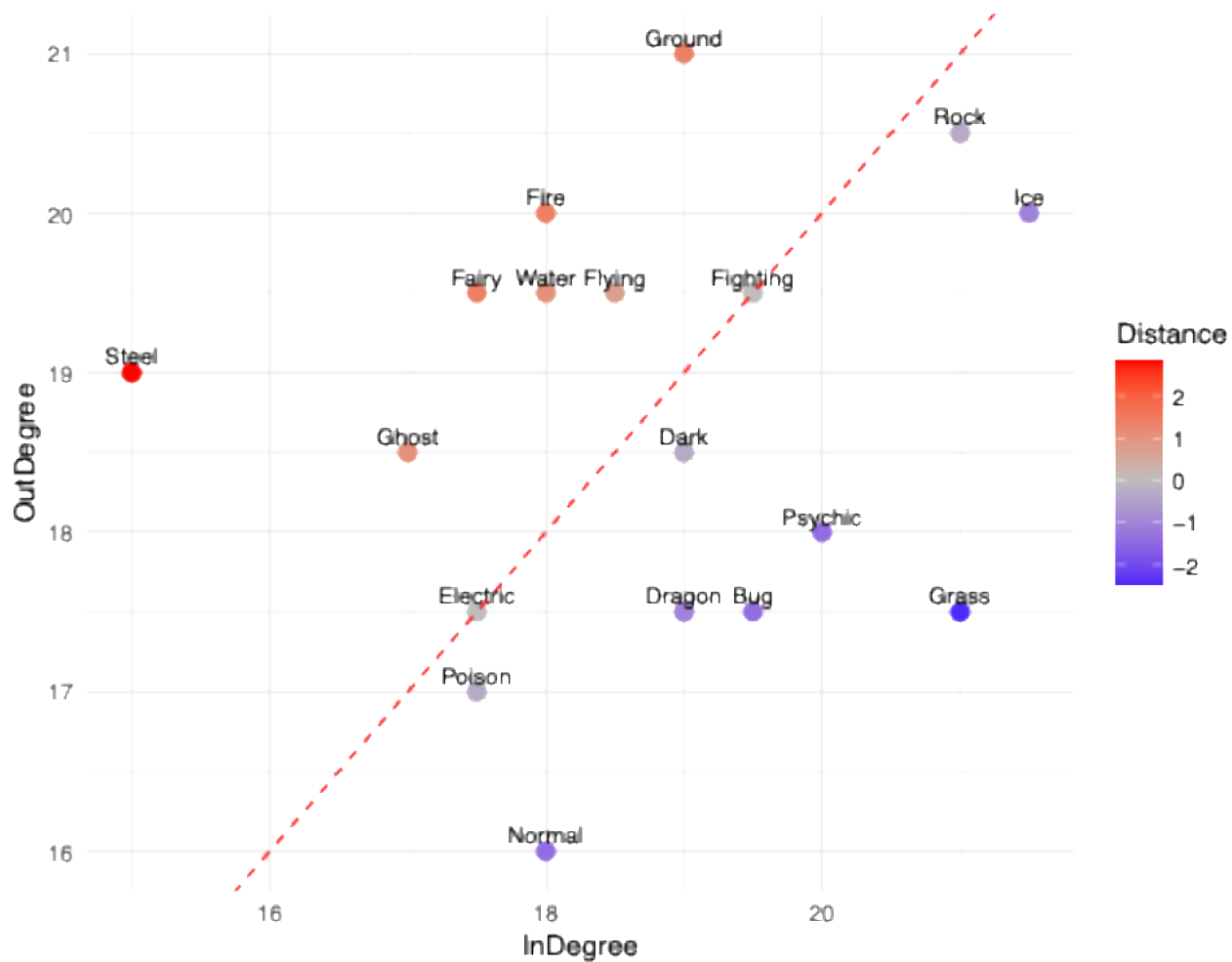


次数中心性の散布図

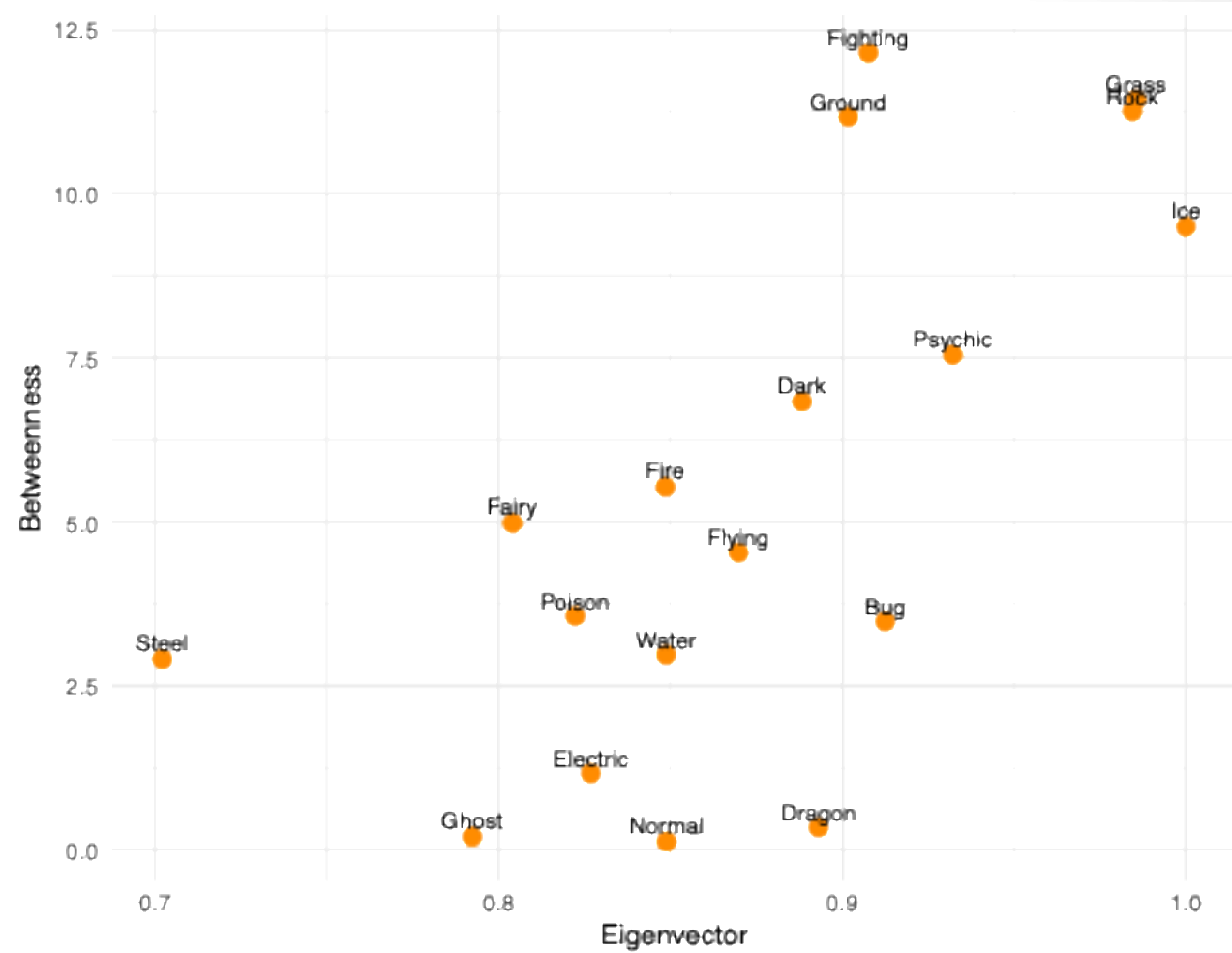


媒介・固有ベクトル中心性の散布図

# 第6世代～での結果



次数中心性の散布図



媒介・固有ベクトル中心性の散布図

# 世代間の比較結果

スコア / 世代	初代	第2～第5世代	第6世代～第9世代
score_degree	0.5777778	0.5630252	0.6111111
score_bet_vec	0.1648264	0.2594863	0.3248465
score_total	0.3713021	0.4112557	0.4679788



世代を経るごとに、  
ゲームバランスが改善されている！



# 今後の展望

- 先ほどの結果について初代→第2～第5世代の次数中心性のスコアのみがなぜ下がってしまったのかを考察する
  - タイプごとに個別で分析する必要がある？
  - 第6世代で「フェアリー」タイプが追加されたのはそれが理由か？
- 新タイプを提案し、新タイプを含めて再実験を行う
  - ゲームバランスが改善されているのになぜ追加する必要があるのか？
  - そのタイプの弱点は？弱点をつけるタイプは？なぜそのように決めた？

# 既存の研究の紹介

- 最も優秀なタイプは？ポケモンのタイプ相性をネットワーク分析する： ※補足⑨

<https://otepipi.hatenablog.com/entry/2019/04/30/134755>

- ポケモンのタイプ相性を可視化する：

<https://qiita.com/s-kosuke/items/a07f4a6bdbd825adc33b>

- ネットワーク分析で見るポケモンランクバトル[S1 シングル]：

<https://www.youtube.com/watch?v=aEW8Z5jbrLE&t=169s>

# 参考文献

テンポラル・ネットワーク

Naoki Masuda, Renaud Lambiotte著 ; 増田直紀訳. --第2版--. 森北出版, 2022.12.

# 補足

## ※①

本プロジェクトでは、「ポケモン」を題材にした内容ですが、他のゲームに関しても 同様の議論ができると考えています。

ポケモンを題材にした理由は、Kaggleでゲーム関連のデータセットを調べた際に、ポケモンに関するデータを発見し、興味を持ったからです。また、データの数も豊富な作品なため、本プロジェクトの内容に限らず、扱いやすい内容だと思いました。

## ※②

ノードの「重要性」は一般的な定義が存在しないため、6つのルールは本プロジェクト内に限った内容です。現に、他の解釈で研究されている方がいらっしゃいます。(補足⑨参照)

# 補足

## ※③

次数中心性のスコアは、次数中心性の散布図から以下のように算出するものとします。

あるノード*i*について、直線 ( $y = x$ ) との距離の偏りは、
$$D_i = \frac{OutDegree_i - InDegree_i}{\sqrt{2}}$$

そして、全体のスコアは、
$$Score_{degree} = 1 - \frac{\text{mean}(|D_i|)}{\text{max}(|D_i|)}$$

上記の計算は、データを0~1の範囲にスケールリングする手法 (Min-Max正規化) を使っています。

また、直線 ( $y = x$ ) は入次数と出次数のバランスを表しています。

つまり、この直線との平均距離が小さい (バランスが良い) ほどスコアは高くなります。

# 補足

## ※④

ネットワーク上での中継点は、攻撃技のタイプから攻撃を受けるタイプに向かう途中に存在します。

つまり、媒介中心性が高いタイプは、「受け(後出し)」の役割がある＝採用されやすい」となります。

ここでいう「受け」というのは次数中心性のように「弱点の数」を考慮して決定されるものではないので、評価基準で区別化ができます。

## ※⑤

固有ベクトル中心性はノードの重要性を「再帰的」に計算して算出します。

つまり、結果として表示される固有ベクトル中心性の値は、ネットワーク全体を考慮した結果になります。

# 補足

※⑥

媒介・固有ベクトル中心性のスコアは以下のように算出するものとします。

媒介中心性と固有ベクトル中心性の相関係数 $r$ を用いて、 $Score_{betvec} = 1 - |r|$

「強い相関がある = 一部のタイプが支配的になる」になるため、相関係数 $r$ は「ゲームバランスの悪さ」を表していると言えます。

よって、相関係数の値が小さいほどスコアは高くなります。

# 補足

## ※⑦

本プロジェクトでは次数中心性と媒介・固有ベクトル中心性のスコアにそれぞれ0.5の重みをかけています。

つまり、**公平性と多様性は重要度を同じ**だと考えています。

もし、片方のみを重要視したい場合には、重みを変える(例えば、0.8と0.2)ことで実現できます。

ここで、次数中心性は**ミクロな指標**(隣接しているノードのみを考慮しているため)、媒介・固有ベクトル中心性は**マクロな指標**(ネットワーク全体の構造を考慮しているため)であることに注意してください。

## ※⑧

初代ポケモンでは、ドラゴンタイプの技は「りゅうのいかり」(固定ダメージ)のみのため、ドラゴンタイプの出次数(攻撃側の倍率)の重みは全て1としてデータセットを作成しました。



# 補足

## ※⑨

特にこの方の内容は、本プロジェクトとタイプの「重要性」の解釈が異なっており、とても参考になるものだったので紹介します。

この方は、タイプの「優秀さ」を次のように定義しています：

「タイプの優秀さ」 = (出次数の合計(タイプの強さ) - 入次数の合計(タイプの弱さ))

これは、弱点をつける数が多く、弱点が少ないタイプが強い(優秀な)タイプであるという指標です。

この指標に基づくと、「はがね」タイプのように明らかに防御よりで設計されたタイプは優秀と判定されます。(結果として、はがねタイプのスコアが最も高かったです)

本プロジェクトでは、確かに次数中心性において「はがね」タイプが最も優秀でしたが、媒介中心性は意外にもかなり低い値となりました。

このように、同じデータでも評価指標が違うと、全く異なる結果が得られます。