九州大学学術情報リポジトリ
Kyushu University Institutional Repository

オンラインゲームにおける不正プレイヤー検出のための関数データ解析による欠損データ復元

張, 忠強
九州大学

川本, 淳平
九州大学

櫻井, 幸一
九州大学

https://hdl.handle.net/2324/1661861

出版情報：暗号と情報セキュリティシンポジウム, pp.3B2-3-, 2016-01-19. 電子情報通信学会情報セキュリティ研究専門委員会(ISEC研)
バージョン：
権利関係：

Copyright© 2016 The Institute of Electronics, Information and Communication Engineers

SCIS 2016 2016 Symposium on Cryptography and Information Security
Kumamoto, Japan, Jan. 19- 22, 2016
The Institute of Electronics, Information and Communication Engineers

オンラインゲームにおける不正プレイヤー検出のための関数データ解析による欠損データ復元

Recover the missing data by using Functional Data Analysis for detecting the illegal players of online game

張 忠強* Zhang Zhongqiang　川本 淳平* Junpei Kawamoto　櫻井 幸一* Kouichi Sakurai

あらまし オンラインゲームは，近年最も人気があるゲームの一つとなっている．しかし，このようなゲームにおいて，ゲームボットの使用やリアルマネートレードなどの不正行為が増加している．不正プレイヤーを検出するために，ゲームサーバーからプレイヤーのデータを取得してプレイヤーの特徴を分析する必要がある．しかし，多人数が同時にアクセスするオンラインゲームのサーバーからゲームデータ取得する時，ある程度の欠損やノイズが発生することがある．オンラインゲームに関するプレイヤーの特徴を分析する時，より高い精度を獲得するために，より正確なデータを求める必要がある．この論文では，World of Warcraft Avatar History (WoWAH)データセットをケーススタディとして，関数データ解析を用いて欠損値を復元する方法を提案する．プレイヤーのゲームプレイング時間と達成度の関係を用いて，曲線の当てはめの方法により，欠損値を予測して，ノイズの少ないデータを推測することができた．そして，処理したデータを使ってプレイヤーの分類を行う．

キーワード オンラインゲーム，不正行為，欠損値，関数データ解析，ノイズ

1 はじめに

オンラインゲームとは、多くの参加者がインターネット上で同時にゲームをすることができるコンピュータゲームの一種である．近年，オンラインゲームは最も人気があるゲームの一つとなっている．しかし，このようなゲームにおいて，ゲームボットやリアルマネートレードなどの不正行為が増加している．これらの不正行為は，通常のプレイヤーにとって邪魔なものであり，ゲームの楽しさを減少させる可能性がある．ゲーム世界のバランスを維持するために，ゲームの運営会社は不正行為をするプレイヤーに対して厳しい対応をとっている. 例えば，ゲームボットを使用するプレイヤーのアカウントを削除することは良く行われる対策の一つである．

ゲームボットはロボット（ROBOT）を語源としている. これらのゲームボットは自動化のプログラムであり，ゲーム内で自動的に行動する．特別な目的を達成するために，ゲームボットは，いくつかの動作を繰り返すように設計される．例えば，ボットが特定な場所を巡り，自動的に敵を倒すことでゲーム内の貨幣を多量に得られる．このような行為によってゲーム内で多量の通貨を生み出されることで, ゲーム内物価が急激に上がることになる．その結果，普通のプレイヤーがゲームをすることを阻害する．また，リアルマネートレードは，ゲーム内で得られたアイテムを現実世界で売買する不正行為である．ゲームの中に, レアアイテムと呼ばれた貴重なものがあり，これらのアイテムは現実世界で数十万円もの金額で取引される場合がある．このような行為が存在すれば，不正なプレイヤーはゲーム内でレアアイテムを取得するために，ゲームボットを使用したり，ゲーム内で詐欺行為が発生したりする．

オンラインゲームにおいて，ゲーム世界のバランスを守るために，様々な規則を定められている．ゲームプレイ時間はという概念は，あらゆるタイプのゲームに存在している．ゲームにおいてレベル（達成度）とゲームプレイ時間がある程度の相関性があると考えられる．プレイヤーがルールに従ってゲームを楽しむ場合, レベル(達成度) とゲームプレイ時間は正の相関関係になっている．しかし，プレイヤーがルールを守らずにゲームをする場合，その正の相関関係を破れる可能性が高くなる．プレ

* 九州大学　819-0395　福岡市西区元岡 744 ウェスト 2-712
zhongqiang.zhang@inf.kyushu-u.ac.jp

イヤーのゲームプレイ時間とレベルを調べることは，プレイヤーがゲーム内で行った行動を理解する有効な方法の一つであると考える．本研究では，プレイヤーのレベルとゲームプレイ時間という二つの特徴量を用いて不正プレイヤーの検知を行う．

一方，プレイヤーの特徴を分析するためには，ゲームからプレイヤーのデータを取得する必要がある．しかし，プレイヤーのデータはハードウェアの環境と電気のノイズ，通信遅延など様々な要因である程度の欠損が起きる．本稿では，データの欠損値に関する問題点について，関数データ解析[6]を使用し補完する手法を提案する．

また，提案手法の評価として，公開されているオンラインゲームのプレイデータの一つである，WoWAH（World of Warcraft Avatar History）を用いた評価実験を行った．このデータには欠損データが存在しているが，提案の関数データ解析を使って，欠損があるデータを予測して，元データを推測することができた．そして，処理したデータを使ってプレイヤーの分類を行う．最後の結果として，不正プレイヤーの検出の精度と再現率を向上させた結果，F1-Measure の値は 0.957 となった．

2　関連研究

オンラインゲームにおける不正行為を検出するために，様々な研究が行われている．前述のように，不正プレイヤーの検出は主にアクションベース，ポジションベース，インプットデータベース，そしてトラフィックベースという４つのカテゴリーに分けられる[1,2,3,4]．

藤田らはゲーム内におけるプレイヤー間の取引量をもとに，Massive Multiplayer Online Role-Playing Games(MMORPG)のネットワークにおけるリアルマネートレードを行うプレイヤーを検出する方法を提案している[1]．Chen らは，ファーストパーソン・シューティングゲーム（First Person Shooter; FPS）においてアバターの移動軌跡を取って，ボット検出のための学習方法を提案している[2]．Gianvecchio らの研究は戦闘ゲームにおいてプレイヤーのキーボードとマウスの入力データを使用し，入力特徴を分析してボットの検出する方法を提案した[3]．

しかし，ゲームの種類は複数あり，それぞれのゲームでは利用できるプレイヤーの特徴が異なっている．従って，上記の研究成果を他のゲームに適用することは難しい．本研究では，ほとんどのゲーム共通なプレイ時間と達成度という特徴量を用いることで，ゲームの種類にとらわれない方法を提案している．

Chen らはゲームサーバーからのトラフィックパケットを分析し，ボットの特徴を発見し，ボットを検出した[4]．

しかし，トラフィックパケットを分析するため，膨大なデータが必要とする．それは，ゲームサーバーに重い負荷を与える．

3　関数データ解析

オンラインゲームにおける不正プレイヤーを検出するために，データからプレイヤーの行動の特徴を分析する必要がある．しかし，データセットに欠損値が存在しているため，データの分析結果に誤り率が高くなる．だから，より精度良く結果を得るために，より正確なデータが必要になる．

関数データ解析(Functional data analyses) は連続変数と関数モデルを分析する手法である．えば，主成分分析，正準相関分析，様々な線形モデルを解析する手法などの方法は，関数データ解析の応用例である．

我々の研究では，プレイヤーの特徴を分析するため，欠損値の少ないデータセットが求められる．しかしながら，実際のデータには多くの欠損値がある．この欠損値を復元するために，関数データ解析を用いる．本論文では，曲線の当てはめにより，欠損値を予測しノイズの少ないデータを推測する手法を提案する．

LOESS アルゴリズムはデータを復元するために，局所重み付け関数を利用することにより，予測するデータを平滑化される方法である．一般的に，オンラインゲームの規則から考えると，プレイヤーのレベルとゲームプレイ時間との関係は正の相関関係にある．すなわち，ある程度の平滑性があると考えられる．このため，本論文では，LOESS アルゴリズムを利用し，より正確なデータを求める．

LOESS アルゴリズムは，局所重み付け関数を使ってデータを平滑化するスムージング手法であり，予測したいポイントの値はスムージングされた隣り合うポイントに影響される．解析するデータの中に，i 番目のポイントを x_i とすると， x_i に対する局所重み付けを計算するとき，LOESS アルゴリズムのプロセスは以下の 4 つのステップとなる[7,8]．

i. まず，インクルードされたポイント n の範囲を選択する．

ii. 式（1）により，各ポイントの局所重み付けを定められる．

$$\mathrm{w}(x_k) = \left(1 - \left|\frac{x_i - x_k}{d_i}\right|^3\right)^3 \qquad (1)$$

式（1）の中に，k の値は 1 から n までであり，d_i はポイント x_i からの第 n 番目までの距離を表す．

iii. パラメータを計算するために，下記の二次関数を最小化する．

$$\hat{y}_k = a + bx_k + cx_k^2 \qquad (2)$$

iv. 式（3）により，ロバスト重み付けを計算する．

式（3）の中に *median* は2つのポイントの中央値を表す．まだ，式から残差が 6 *median* 以上になる場合のロバスト重み付けは 0 にあることがわかる．

$$G(x_k)\begin{cases}\left(1-\left(\dfrac{\left|y_i-\hat{y}_i\right|}{6median\left(\left|y_i-\hat{y}_i\right|\right)}\right)^2\right)^2 & \left|\dfrac{\left|y_i-\hat{y}_i\right|}{6median\left(\left|y_i-\hat{y}_i\right|\right)}\right|<1\\ 0 & \left|\dfrac{\left|y_i-\hat{y}_i\right|}{6median\left(\left|y_i-\hat{y}_i\right|\right)}\right|\geq 1\end{cases} \quad (3)$$

最終的な式は以下のようになり：

$$\sum_k w(x_k)G(x_k)\left(y_k-a-bx_k-cx_k^2\right)^2 \quad (4)$$

式（4）の中に，$W(x_k)$ と $G(x_k)$ はそれぞれ局所重み付け関数とロバスト重み付け関数を表す．期待される近似曲線が得られるまで，このプロセスを継続して実行される．

4　不正プレイヤー検出への特徴ベクトル

本節では，不正プレイヤーを検出するプロセスの概要をはじめに説明し，その後特徴ベクトルの設計と LOESS アルゴリズムを用いた処理を順に説明する．

4.1　不正プレイヤーを検出するプロセス

まず，オンラインゲームにおける不正プレイヤーの検出するモデルを紹介する．

図１　不正プレイヤーを検出するプロセス

図１には，不正プレイヤーを検出するための一般的なプロセスを示している．所有したデータセットから，不正プレイヤーの行動を分析し，不正行為を分けるような特徴を抽出し，特徴集合を生成する．次では，生成した特徴集を処理して特徴ベクトルを生成する．これらは，データの前処理と呼ばれる．このようにして前処理を施した特徴ベクトルを分類フィルタにかけて不正プレイヤーの検出を行う．

4.2　特徴ベクトルの設計

本研究では，ゲームプレイ時間と達成度であるレベルを特徴量として扱う．解析対象となるデータに含まれる総プレイヤー数が N とするとデータセットは下記のように書ける．

$$X=\{x_m \mid m=1,2,\ldots,N\} \quad (5)$$

式（5）おいて，x_m は各プレイヤーの特徴ベクトルを表し，$x_i=(x_{i1},x_{i2},\ldots x_{in})$ と書ける．i 番目のプレイヤーに対応する特徴ベクトル x_i 中の x_{ij} は，そのプレイヤーがレベル j を達成するのに要したゲームプレイ時間を表している．例えば，$x_1=(2,13,\ldots 111)$ の場合，値 13 は，プレイヤー x_1 がレベル 2 を達成するのに要したゲームプレイ時間を表す．

4.3　LOESS によるデータの処理

上記のように設定した特徴ベクトルにおいて，欠損値を存在している．具体的には，ある xij の値が記録されていないという問題である．我々は，LOESS アルゴリズムを用いて，関数モデルを分析し，データの曲線フィッティングを生成することでこの欠損値を補完する．

LOESS アルゴリズムを実現するために，今回は R 言語をツールとして使って，データの解析を行った Algorithm 1　は LOESS アルゴリズムを応用するためのインタフェイスのコーディングである．R 言語の LOESS 関数を利用するために，まずはデータセットを導入する必要がある．R 言語の導入関数にふさわしいデータ構造を扱うため，各プレイヤーのデータを 1 つのファイルに書き込む．まずは，setwd にそれらのファイルのパス入力し，各プレイヤーの入力したデータをプログラムに導入する．次に，プログラムの実行するスペース

Algorithm 1 Interface by using LOESS

```
setwd("FILE PATH")
a = list.files
dir= paste("WORK SPACE")
n = length(dir)
for (i in 1:n) do
  z<-0
te<-read.table("INPUT FILE")
model1=LOESS("INPUT DATA")
png(paste("OUTPUT IMAGE")
  for (j in 1:60) do
  z<-c(z,predict)
write.csv("OUTPUT DATA")
  end for
end for
```

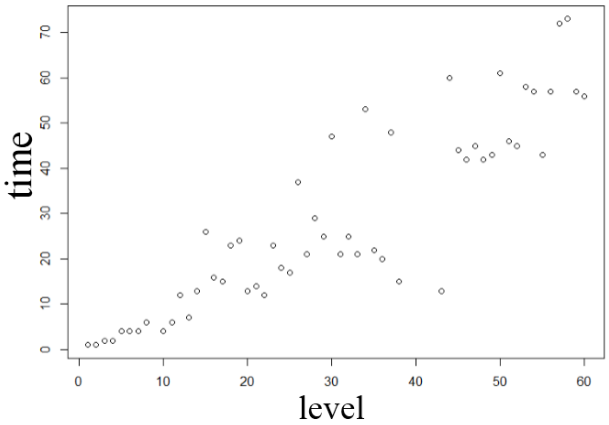

図２　各レベルでのゲームプレイ時間の分布

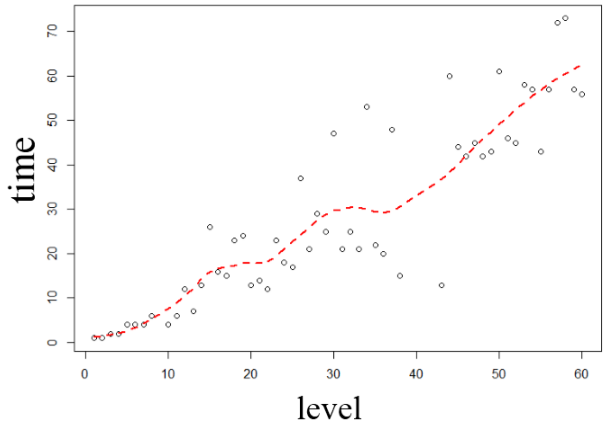

図３　LOESS 法による模擬した曲線

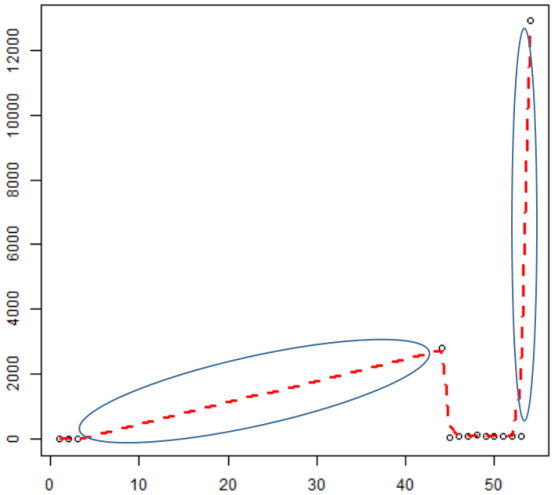

図４　大きい欠損値が存在した曲線フィッティング

を設定し，実行した結果をそのスペースに保存する．入力したファイル量が終わるまで，各ファイルを LOESS 関数で処理を行い，そして，レベル１から６０までのデータを解析した関数モデルで推測する．最後に，出力した結果を保存し，プログラムを終わる．

Algorithm 2　K－means アルゴリズム

```
Input:
  Users: the object of the players
Output:
  Marked users: the users is marked by labels
1: users = readObject ();
2: ClusterPoint= randomSelect ();
3:   for distance < threshold do
4:     for   user in the users   do
5:       for   point in the ClusterPoint   do
6:           find the nearest point;
7:           mark the user;
8:                 end for
9:         end for
10:   ClusterPoint = getNewPoint ();
11:   calculate distance;
12:   end for
13:   return users
```

図２には，あるプレイヤーのゲームデータであり，レベルを横軸とし，ゲームプレイ時間を縦軸とし，散布図を生成する．図２から，データの分布は分散であり，ある程度の欠損値が存在していると考える．

図３には，R 言語の LOESS 関数で処理した結果を示している．赤い曲線は解析した関数モデルの曲線であり，各レベルでのゲームプレイ時間はこの得られた曲線で再計算する．このようにして，レベルとゲームプレイ時間の特徴ベクトルが収束される．

まだ，図4のような非常に大きな欠損値が存在しているサンプルがあった．このようなデータを処理する時，LOESS アルゴリズムにより，大きな欠損値があるところは1次関数モデルで処理する．この原因で，我々が１次関数モデルを使ってデータの処理を行い，LOESS 関数との結果と比較した．比較した結果は後ろのセクションで説明する．

5　k-means アルゴリズム

データの前処理を行った後，不正プレイヤーを検出するために，今研究では，k-means アルゴリズムを用いてプレイヤーの分類を行う．

表 1　データセットのサンプル

Query Time	Seq.#	Avatar ID	Guild	Level	Race	Class	Zone
01/01/06 23:59:39	1	467		1	Orc	Warrior	Orgrimmar
01/01/06 23:59:39	1	921	19	1	Orc	Shaman	Orgrimmar
01/02/06 00:03:31	45	1367	8	60	Undead	Warrior	Arashi Mountain

表 2　データセットの概要

Start date	2006-01-01
End date	2009-01-10
Duration	1,107 days
Sampling rate	144 samples per day
# of samples	159,408
# of missing samples	21,324
# of avatars	91,065
# of sessions	667,032

k-means アルゴリズムは距離ベースのクラスタリング手法であり，オブジェクトの距離を類似性の評価指標として使用する[9]．各サンプルが一番近くのクラスタセンターに分けられる．

本セクションでは，k-means アルゴリズムのプロセスと類似度の計算する手法を説明する．

5.1　k-means アルゴリズムのプロセス

Algorithm 2 は k-means アルゴリズムの概略を示している．前処理を行ったプレイヤーのデータを入力するデータとして扱い，k-means アルゴリズムにより，分類したクラスタをマークして出力する．まず，入力するデータを初期化してランダムで k 個のクラスタセンターを選択する．次に，入力したオブジェクトが最も類似したクラスタセンターに割り当てられて，k 個のクラスタを生成する．そして，各クラスタの中心を計算して新しいクラスタセンターとして使って各オブジェクトを再分類する．クラスタセンターの移動距離が収束するまで，このプロセスを繰り返される．

1) ランダムで中心ベクトルを選択する．
2) クラスタに割り当てられる．
   - 各サンプルが一番近いクラスタセンターに分類する．
3) 各クラスタの中心を計算する．
   - 分類されたクラスタの中に中心ベクトルを計算する
4) クラスタセンターの移動距離が収束するまで，2 と 3 ステップを繰り返す．

5.2　ユークリッド距離

k-means アルゴリズムを使うために，オブジェクトの間の類似度を計算する必要がある．実験には，最も一般的に使われるユークリッド距離を用いてオブジェクトの間の類似度を計算する．

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \qquad (6)$$

2 つのオブジェクトの間の類似度が d(xi, ,xj) で計算する．d(xi, ,xj) が距離の意味をしているので，d(xi, ,xj) が小さくなるなら，サンプルとの間は大きい程度で類似している．逆に d(xi, ,xj) が大きくなると，サンプルとの間は小さい程度で類似している．

6　実験と結果の評価

本稿では，WoWAH（World of Warcraft Avatar History）のデータを用いてオンラインゲームにおける不正プレイヤーを検出する手法を提案した．WoWAH は最も人気がある MMORPG(Massively Multiplayer Online Role-Playing Game)の 1 つであり，12 万人以上のプレイヤーが持っている．WoWAH のデータセットはインターネットで公開された．

6.1　データセットの概略

表 1 には，データセットのサンプルを示している．表 2 から，データセットの中は 8 つの属性があることがわかる．それらはサンプリングタイム，サンプリングシリアルナンバー，アバターID，ギルド，レベル，種族，クラス，ゾーンである．

表 1 に各属性はサンプリングされた時，記録されたデータであり，今回の研究では，その中プレイヤーのアバターID を検索し，各アバターID の各レベルでのゲームプレイ時間を取得して計算する．そして，プレイヤーのレベルとゲームプレイ時間をセットして特徴ベクトルを設計する．

表 2 には，WoWAH データセットの概略を示している．このデータセットは 2006 年から 2009 年までの 3 年間のデータであり，サンプリング時間は毎 10 分間 1 回である．表 1 から，91065 のアバターID を記録され，サンプル総量は 159408 である．その中に，21324 の欠損したサンプルが含まれた．

6.2　クラスタリングによるプレイヤーの分類

今研究のデータセットは 2 つに分けられた．あるデータに非常に大きな欠損値が存在しているため，プレイヤ

ーの特徴分析に大きな誤差を起こし，実験に使えないとなる．データセット１は，そのような大きな欠損値があるサンプルを取り除き，一千人分のデータ集合を集まる．その中に，３２人の不正プレイヤーがマークされた．データ２は，図４のような大きな欠損値があるサンプルを含んでいる１２５７人のデータ集合集まり，その中に 46 人の不正プレイヤーがマークされた．

プレイヤーのレベルとゲームプレイ時間の相関関係を説明するために，我々がすべてのプレイヤーのデータを用いて，全プレイヤーの平均的なレベルとゲームプレイ時間の水準を求めた．

図５に，横軸はプレイヤーのレベルであり，縦軸は全プレイヤーの各レベルでのゲームプレイ時間である．曲線の形から見ると，プレイヤーが各レベルでのゲームプレイ時間単位は指数関数的増えることがわかる．ゲームの中に，不正プレイヤーの数が少ないので，全プレイヤーの平均的な水準を評価標準として使え，不正プレイヤーの発見に役に立つと考えられる．

プレイヤーのレベルとゲームプレイ時間をセットして特徴ベクトルを生成した後，k-means アルゴリズムにより，プレイヤーの分類を行う．本稿では，2 つのデータセットを用いて関数データ解析により，データの前処理をし，別の前処理の手法と比較する．

WoWAH のデータセットの中に，プレイヤーの最高のレベルは８０である．しかし，ゲームの運営の間にゲームバージョンを更新したことがあった．このため，最高のレベルが６０を８０に変更した．この原因で，６０後のレベルとその前のレベルとの相関性が破れ，使えなくなる．まだ，データの中に欠損値を復元する時，６０後のレベルでのゲームプレイ時間が前のデータに影響をさせる可能性があるため，レベルが６０までのデータを使用した．

関数データ解析の有効性を説明するために，我々がデータセット１を用いて，３つの手法で実験を行う．それぞれは単独の k-means アルゴリズム法，Latent Dirichlet Allocation(LDA)アルゴリズム[10,11]，関数データ解析である．単独の k-means アルゴリズム法は，データの前処理を行わず，元データから特徴ベクトルを抽出し，プレイヤーの分類を行う．LDA アルゴリズムは文章をモデリング化する手法であり，生成するトピックの分布確率を求めるアルゴリズムである．今回では，LDA アルゴリズムの詳しい説明を行わずに，実験の結果だけを示す．関数データ解析手法により，タイプが類似しているプレイヤーの特徴が収束することができる．即ち，タイプが類似しているプレイヤーの特徴が著しくされ，プレイヤーの分類はよりしやすくなる．

今回の実験では，k-means アルゴリズムのパラメータ k は２から１２までを実験した．普通なプレイヤーを１つグループに割り当てられ，不正なプレイヤーを他のグループに割り当てられている．３つの方法で不正なプレイヤーを検出する精度と再現率は表３に示している．

表３の中に，P(Precision rate) と R(Recalling rate) はそれぞれ正確率とリコール率と表す．第一列は k-means アルゴリズムのパラメータ k の値であり，後ろの３列はそれに対する３つの方法での正確率とリコール率の値である．それぞれは単独の k-means アルゴリズム法，Latent Dirichlet Allocation(LDA)アルゴリズム，関数データ解析の方法である．まだ，実験結果への評価にこれらの結果を総合的な評価標準の F1-Measure の値で評価する．

データセット２の中に，大きな欠損値が存在しているので，本稿では，関数データ解析により，欠損値を復元し，k-means アルゴリズムを用いてプレイヤーの分類を行う．R 言語の LOESS では，図４のようなデータを復元する時，非常に大きな欠損値があるところは１次関数モデルで処理する．このため，我々が１次関数モデルを使って，データの前処理を行ってプレイヤーの分類を行う．そして，得られた結果と LOESS アルゴリズムで処理した結果を比較する．

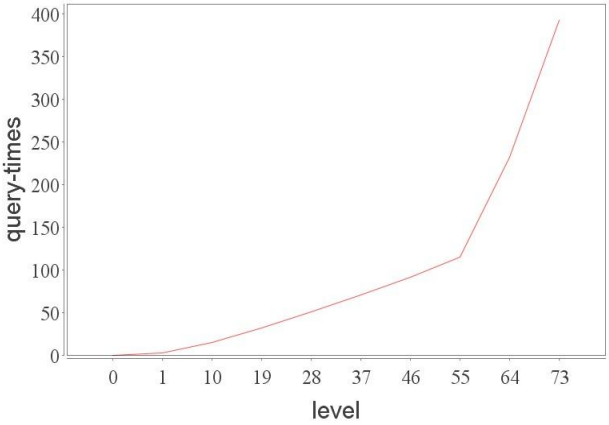

図５　全プレイヤーの平均的なん水準

表３　正確率とリコール率

k	k-m(P , R)	LDA(P , R)	FDA(P , R)
2	1 , 0.031	1, 0.093	1, 0.03
3	1, 0.391	1, 0.375	1, 0.31
4	0.828, 0.781	0.9, 0.843	0.93, 0.84
5	0.828, 0.906	0.909, 0.937	0.93, 0.906
6	0.828, 0.781	0.9, 0.843	0.89, 0.781
7	0.828, 0.906	0.909, 0.937	0.909, 0.937
8	0.784, 0.906	0.811, 0.937	0.909, 0. 937
9	0.737, 0.875	0.604, 0.937	0.882, 0. 937
10	0.612, 0.937	0.419, 0.968	0.6590.963
11	0.438, 1	0.418, 0.968	0.508,1
12	0.417,1	0.361, 0.968	0.492,1

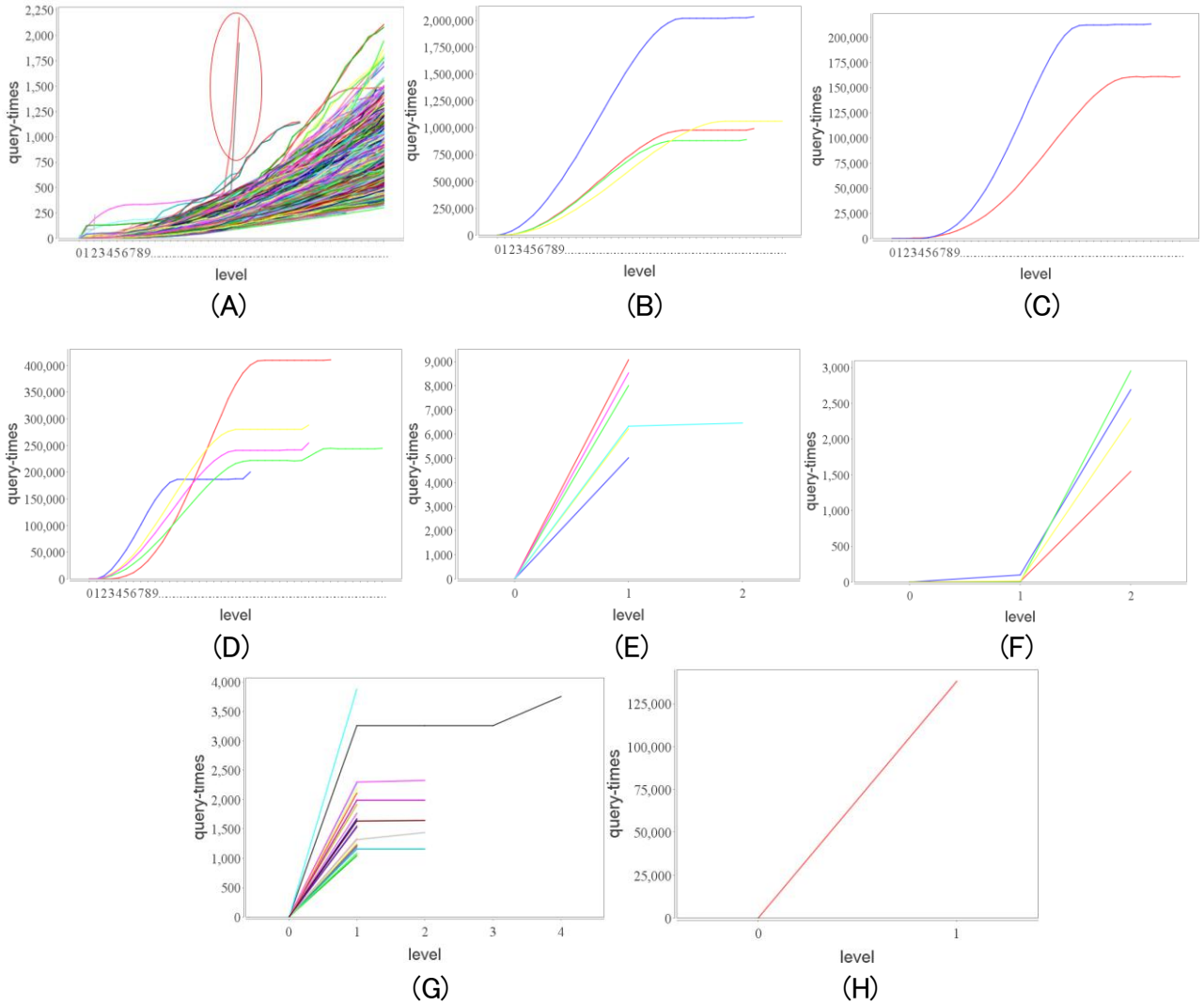

図６　FDA を用いてデータを処理したプレイヤーの分類(k=8)

この実験に，k は２から１２までを設定し，分類を行う．パラメータ k の値が結果に大きく影響するので，本稿では，その中に最も良い分類した結果を示してプレイヤーのレベルとゲームプレイ時間の相関関係を説明する．k ＝ 8 の時，図６のような８つのクラスタを得ることができた．全プレイヤーの平均水準と比較し，クラスタ A のようなプレイヤーはレベルでの必要なゲームプレイ時間単位も指数関数的に増えることがわかり，これらのプレイヤーは普通なプレイヤーだと判別することができる．しかし，クラスタ A に赤い丸のような誤判断された２つの不正なプレイヤーも存在している．この二人のプレイヤーの特徴ベクトルが普通なプレイヤーと異なるので，別のクラスタに分類すべきだと判断する．まで，クラスタ B，C，D の特徴が類似しているが，レベルでゲームプレイ時間総量が非常に大きくなり，不正なプレイヤーと判断された．クラスタ E，F，G，H のようなプレイヤーがレベル１あるいはレベル２でゲームプレイ時間量は千万以上の単位に達し，普通なプレイヤーではそんなに不正常なことがしないため，不正なプレイヤーだと判断した．

表４には，関数データ解析でデータの前処理を行ってプレイヤーを分類した結果の詳細を示している．４４の不正なプレイヤーを検出したが，二人の不正が普通なプレイヤーと誤判断された．そして，二人の普通なプレイヤーが不正に誤判断された．

表４　FDA で処理したデータを用いた分類の結果

	普通	不正
普通（実験）	1209	2
不正（実験）	2	44

6.3　実験結果への評価

最後に，本稿では F1-Measure を用いて実験の結果を評価する．F1-Measure では，結果の正確率とリコール率を組み合わせる総合的な評価標準であり，計算する式は以下のようになる．

$$F1-Meansure=\frac{2*\mathrm{Re}call*\mathrm{Pr}ecision}{\mathrm{Re}call+\mathrm{Pr}ecision} \quad (7)$$

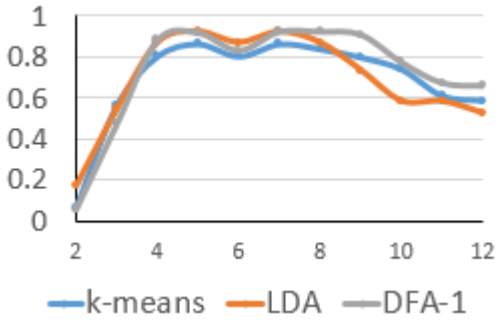

図７　k によるデータセット１の F1-Measure

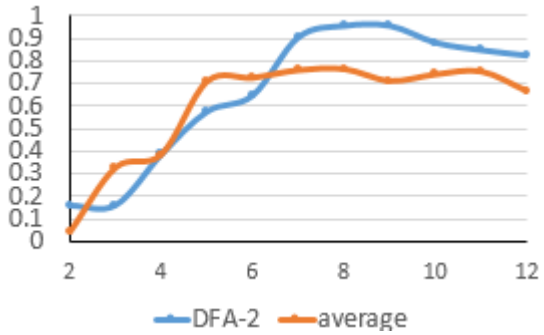

図８　k によるデータセット２の F1-Measure

図７に, k=7 の時 F1-Measure は 0.923 に達した ; LDA 法で，プレイヤーの特徴を著しくなり，k は大きくなると, 誤り率を大きくなる. 図８に, k=8 の時 F1-Measure は 0.957 に達した ; FDA 法で，プレイヤーの特徴を収束して，k は大きくなると，異なる特徴の分類しやすくなる.

7　まとめと今後の課題

本稿は，WoWAH データセットにおける不正な行動を行うユーザーを検出することを目標として，関数データ解析の手法を用いてデータの欠損値を復元した. k-means アルゴリズムを用いて，プレイヤーのゲームプレイ時間に基づく特徴ベクトルを生成し, 不正行動の分類を行い, 特異な行動を持つプレーヤーグループを検出した．そして，他の手法と比較し，関数データ解析はより高い正確率を得ることができた.

今後の課題では，別のタイプのゲームデータを用いて提案した手法の適用性を検証する.

参考文献

[1] Fujita, Atsushi, Hiroshi Itsuki, and Hitoshi Matsubara. "Detecting Real Money Traders in MMORPG by Using Trading Network." AIIDE. 2011.

[2] Chen, Kuan-Ta, Hsing-Kuo Kenneth Pao, and Hong-Chung Chang. "Game bot identification based on manifold learning." Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games. ACM, 2008.

[3] Gianvecchio, Steven, et al. "Battle of botcraft: fighting bots in online games with human observational proofs." Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[4] Chen, Kuan-Ta, et al. "Identifying MMORPG bots: A traffic analysis approach." EURASIP Journal on Advances in Signal Processing 2009 (2009): 3.

[5] Yeng-Ting Lee, Kuan-Ta Chen, Yun-Maw Cheng, and Chin-Laung Lei, "World of Warcraft Avatar History Dataset," In Proc. of ACM Multimedia Systems 2011, Feb 2011.

[6] http://mmnet.iis.sinica.edu.tw/dl/wowah/

[7] Ramsay, James O. Functional data analysis. John Wiley & Sons, Inc., 2006.

[8]https://www.hulinks.co.jp/support/flexpro/v7/dataanalysis_LOESS.html

[9] Wagstaff, Kiri, et al. "Constrained k-means clustering with background knowledge." ICML. Vol. 1. 2001.

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation," the Journal of machine Learning research 3 (2003): 993-1022.

[11] Zhongqiang Zhang, Hiroaki Anada, Junpei Kawamoto and Kouichi Sakurai. "Detection of Illegal Players in Massively Multiplayer Online Role Playing Game by Classification Algorithms." Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on. IEEE, 2015.