# minMax 法により最善手を打つプログラムを用いた局面の変化点を分析する

## 【重要】以下の欄をすべて埋めてから応募してください

氏名：[redacted]

所属学部学科(専攻)：[redacted]

email：[redacted]

【考察記述欄】作業をすべて終えてから考察を自由に記述してください。実行結果の考察だけでなく、データを集めたりプログラムを実行するにあたって苦労した点や、こんなデータを収集、解析してみたい、プログラムを改造してこんな解析や予測もしてみたいというアイデアなどでも構いません。また、実際にプログラムを改造するなど特別に工夫したことがあれば記載してください。

- - - 記述欄はじめ

データを分析した結果、グラフ 1 に示した、経過時間の割合と評価値の関係については、先手である人間が有利なように手を打つため、前半においてほとんどすべて人間が有利である評価値が算出されていた。しかし中盤から後半にかけては、分布が多様化しており一概に結論を出すことが難しかった。参考として 4 次の近似曲線を赤線で表示させたが、それによると中盤あたりからコンピュータが優勢となり後半にかけてもおおよそその傾向が続くと分析できる。しかし、先にも述べたように広い範囲に評価値が分布しており、近似するには無理があるデータであると感じた。統計学では決定係数 $R^2$ を用いることで回帰分析の結果の精度を確かめることができるが、MATLAB の言語を触ったことがなかったため、計算・実装することができなかった。

また、グラフ 2 の経過時間の割合と minMax 法の戻り値の関係についてであるが progress が 0.3 から 0.5 で極端なデータが出ているときは、minMax 法で勝利または敗北のどちらかに決定すると判断しているもので、結果的にそれは間違いであったことがわかる。したがって、今回の評価関数と深さ 6 の minMax 法では人間の指す手を読み切れず、勝利または敗北すると誤認してしまっていることが分かる。深さを 7、8 と深くしていくと計算時間が指数関数的に増加するため現実的ではない。そこで、評価関数を改善する必要があることが分かった。

最後に、グラフ 3 のヒストグラムであるが、このヒストグラムから先手が指す最初の手で評価値が上がることから 0 からおよそ 270 の評価値の階級が最も多くなると分かる。この階級を頂点として左右対称に度数が下がっていくが、階級値-750 あたりで再び増加しており、全体として見ると評価値がマイナス(コンピュータ有利)であることがわかる。これは勝敗の結果、3 勝 7 敗と関係があると考えられる。

今回の分析を通して、自作のプログラムを使ったため、minMax 法の実装や csv の書き出し機能を実装することがとても苦労した。さらに MATLAB 言語を使った経験がなかったため、URL にあった公式チュートリアルを参考に学んだ。コードを実際に書きながら学ぶことができるチュートリアルでとても分かりやすかった。サンプルコードを一切使わずに自力でプログラムを書くことができたので、とても達成感があった。しかし、評価値と経過時間の割合の関係の説明がうまくいかなかったのは残念である。とはいえ、最初は経過ターン数と評価値の関係を見ようとしていたため、試合によって試合終了までに要したターン数が違うことから同条件での比較ができず困っていたが、終了ターン数までの割合(経過時間の割合と表現している)を考えるといった発

想が出てきたときは達成感があった。csv として書き出したものの使用していないデータもあるので時間があれば分析を行ってみたい。

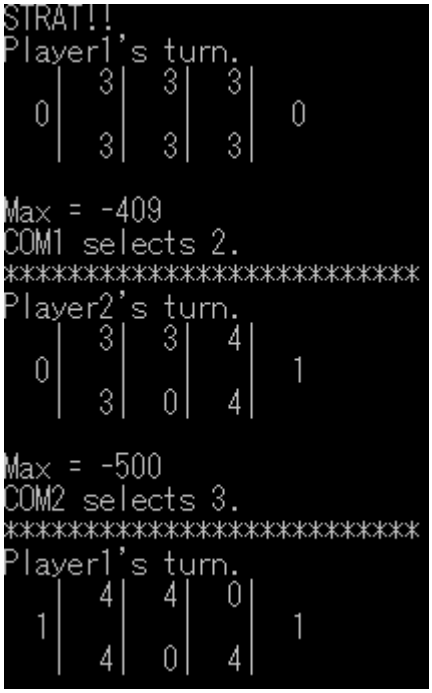
記述欄おわり---

(ここからプログラムが始まります)

## 事前準備:

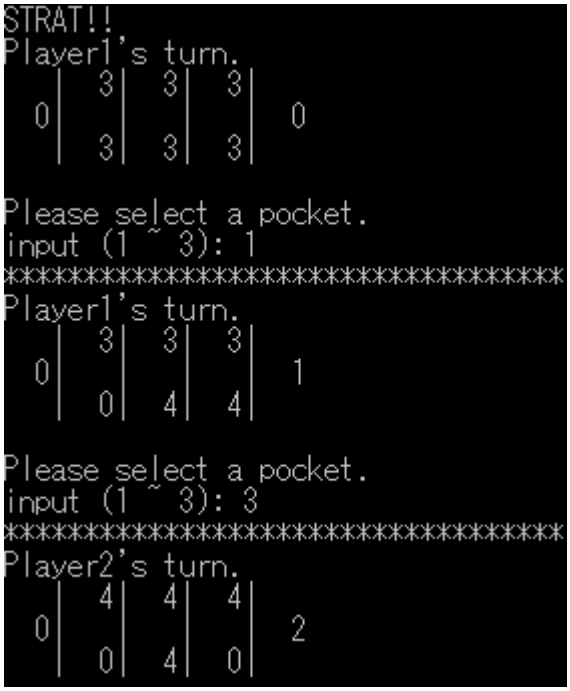
- C++で Mancala ベーシックを minMax 法で分析・対戦できるプログラムを作成する
- Mancala プログラムに、csv 形式での出力機能を搭載する。
- Mancala プログラムを 100 回プレイし、対戦の履歴情報を csv 形式にまとめる。

## マンカラ・ベーシックのルール:




上の画像を用いてマンカラ・ベーシックのルールを示す。まず、これは COM1 対 COM2 の対戦であり、COM1(実験では人間)は画面向かって下側の 3 つの枠、COM2(実験では COM)は上側の 3 つの枠が陣地である。両脇にあるポケットはゴールという。自分のターンが来たときは、選んだポケットの石をすべて手に取り、反時計回りに選んだポケットの隣のポケットから 1 つずつ石を置いていく。「COM1 selects 2.」の 2 は COM1 から見て左から 2 番目のポケットを選んだことを示している。

ゴールのポケットに、移動する最後の石を置いた場合、下の画像のようにもう一度自分の陣地の石を動かすことができる。(プログラムの性質上、各ターン、石を動かす前の状態が表示されている。)




このようにして、自分と相手とが交互に石を動かしていき、自分の陣地の石が最初に 0 個になったプレイヤーが勝ちというルールでゲームを行うのがマンカラ・ベーシックである。

補足として説明するが、「Max = -490」の表示であるが、これは minMax 法を実行したときに帰ってくるコンピュータから見た評価値である。

## Mancala プログラムの説明:

今回の実験に用いる Mancala プログラムの説明を記す。

Mancala のルールはたくさん存在するが、今回は先に述べたマンカラ・ベーシックのルールにのっとり 3 つの石と 3 つのポケットでゲームを行うプログラムを作成した。今回の実験で重要になる譜面の評価関数であるが、

$$f(player, i) = (player\text{の人間から見たときに左から}i\text{番目の石の数})$$

$$g(player) = (\text{人間から見て左から何番目のポケットに初めて石が置かれているか})$$

とし、

$$score = \left(100 \times \sum_{i=0}^{3} f(人, i) - 10^{g(人)}\right) - \left(100 \times \sum_{i=0}^{3} f(COM, i) - 10^{g(COM)}\right)$$

を評価値として計算した。この評価関数を定義するにあたり、相手よりも石が少ないこと、早く石を自分のポケットからなくすことができる見込みが重視されるように設計した。スマートフォンアプリとしてリリースされているマンカラ・ベーシックがプレイできるアプリの一定の難易度を相手にしてもほぼ確実に勝利できることが確かめられている。これよりも適した評価関数も考えられることは、重々承知であるが今回はこの評価関数の問題点を明らかにするという意味でもこれを評価関数として用いることにする。

minMax 法において何手先までを探索するかを任意の値に指定することができるようなプログラムになっているが、今回は深さ 6 の探索を行うこととする。

出力する csv ファイルは、

progress, count, PLAYER, Addr, ALLdata, score, score - pre_score, Winner

といった形式にした。各項の説明を行う。

- progress :試合が終わりのターン数に対して現在のターン数の割合を示す(20 ターンで試合が終了したとき 10 ターン目のデータの場合は 0.5)
- count :現在のターン数
- PLAYER :操作したプレイヤー(1=人間,2=COM)
- Addr :石を動かしたポケットの番地(人間から見て左側から 1,2,3)
- ALLdata :操作後のポケットのデータを格納
- score :評価値
- score - prescore:評価値の変化量
- Winner :ゲームの勝者(1=人間,2=COM)

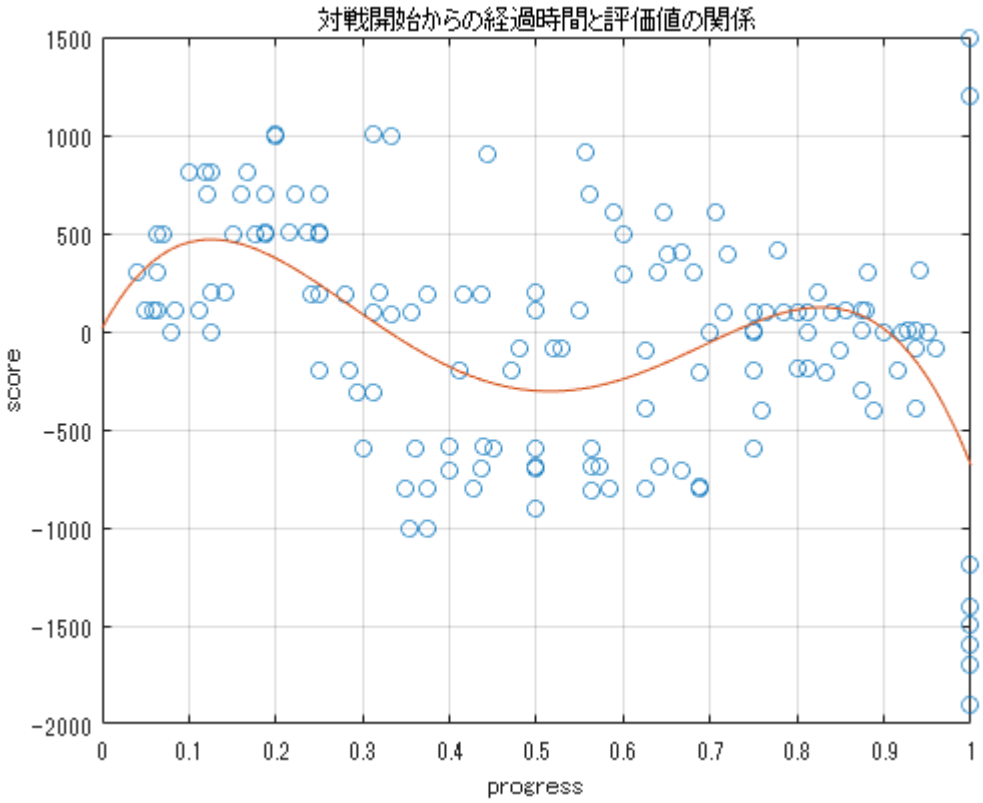
## ファイルからデータを読み込む

log.csv をダブルクリックして、インポートツールを表示し、データを列ベクトルとしてインポートする。

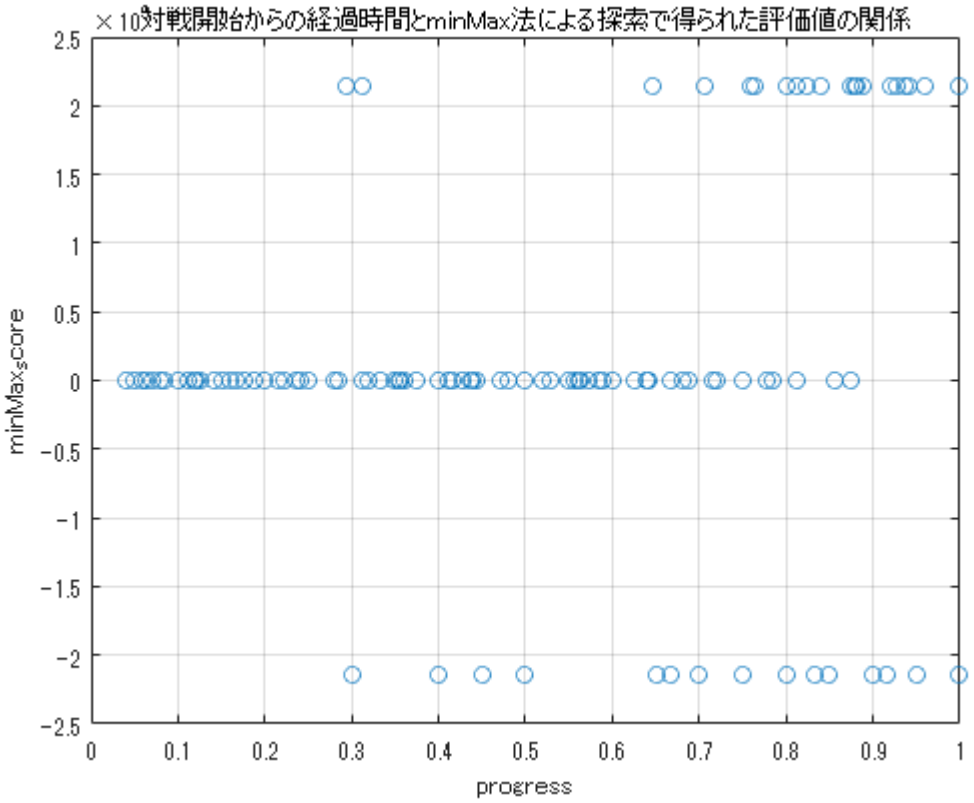
## プログラムを実行してグラフを確認

### グラフに出力(経過時間(割合)と評価値のグラフ)

```
figure(1)
x2 = 0:.01:1;%0.1 刻みで次数 4 の近似曲線をプロットするためのベクトル
p = polyfit(progress, score, 4);
y2 = polyval(p, x2);
plot(progress, score, 'o', x2, y2)
grid on%グリッドをオン
xlabel('progress')%x 軸のラベル
ylabel('score')%y 軸のラベル
title('対戦開始からの経過時間と評価値の関係')
```



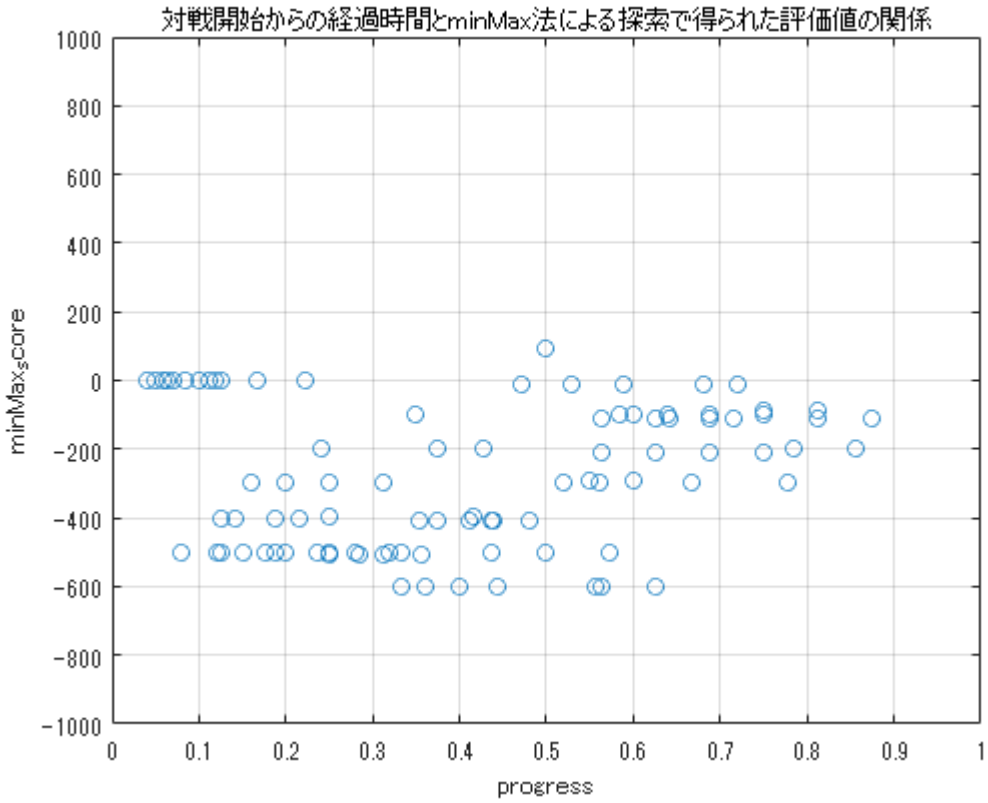

```
figure(2)
plot(progress, minMax_score, 'o')
grid on%グリッドをオン
xlabel('progress')%x 軸のラベル
ylabel('minMax_score')%y 軸のラベル
title('対戦開始からの経過時間と minMax 法による探索で得られた評価値の関係')
```




上記の表は、minMax 法による探索で得られた最善手の評価値と対戦開始からの経過時間をプロットしたものである。値が極端に大きかったり、小さかったりするのは、プログラムで記述した minMax 法が探索した際、最終的に得られた手で COM が勝利できる場合は int 型の最大数-1、COM が敗北する場合は int 型の最小値+1 を返すようにしているからである。

したがって、極端な値を除いたデータを下にグラフ化する。

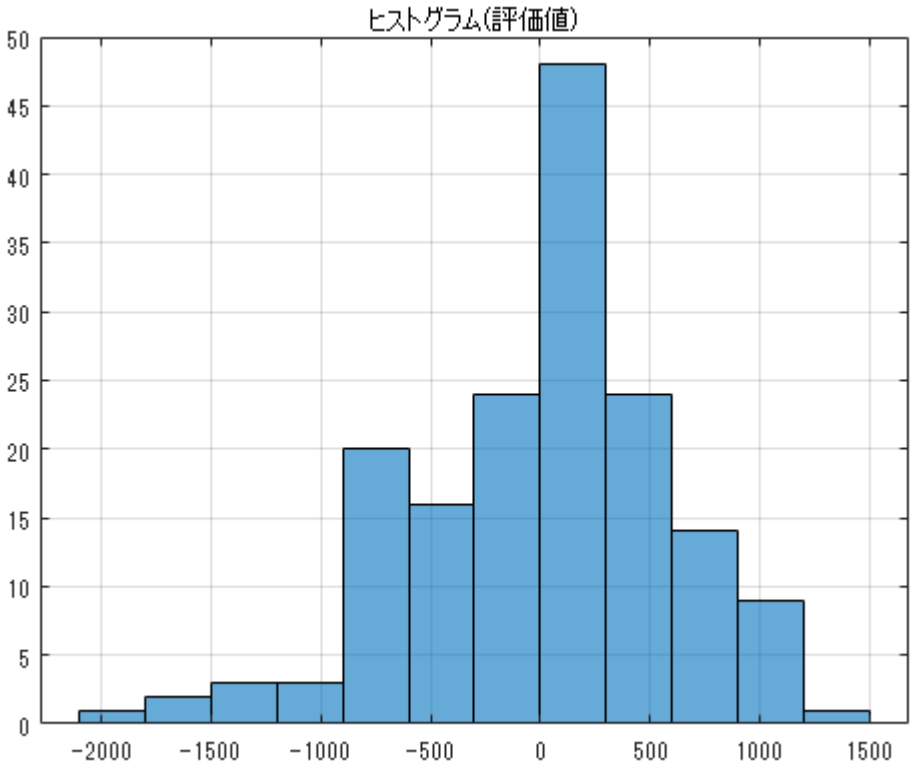
```
figure(3)
x2 = 0:.01:1;%0.1 刻みで次数 5 の近似曲線をプロットするためのベクトル
p = polyfit(progress, minMax_score, 5);
y2 = polyval(p, x2);
plot(progress, minMax_score, 'o')
grid on%グリッドをオン
xlabel('progress')%x 軸のラベル
ylabel('minMax_score')%y 軸のラベル
title('対戦開始からの経過時間と minMax 法による探索で得られた評価値の関係')
axis([0 1 -1000 1000])
```




## グラフに出力(ヒストグラム)

次はヒストグラムを作成してみましょう。

```
figure(4)
histogram(score) %最高気温のヒストグラムを赤で表示
grid on
title('ヒストグラム(評価値)')
```




## 考察:

実行結果に対して、冒頭の考察記述欄に考察を自由に記述してみましょう。

## 保存(レポート化):

ここまでの作業お疲れさまでした。最後に作業の結果を残しておきましょう。

このプログラムは実行結果とともにレポート出力できます。

[ライブエディター]タブの[エクスポート]ボタンを押して、PDF を選んでエクスポートしてください。

保存された PDF 形式のファイルをコンテスト窓口に送付してください。