

# 情報工学実験Ⅱレポート（探索アルゴリズム1）

曜日 & グループ番号: 月曜日 & グループ 2

2015 年 12 月 11 日

## 概 要

この骨組み（テンプレート）を利用する際には、不要な箇所を削除した上で提出すること。例えばこの要旨やコメント文の殆どは「当間から学生へのコメント」であって、「課題に対するレポート（報告書）」ではない。

このレポート（ファイル）は、「情報工学実験Ⅱ・探索アルゴリズムその 1[1]」の実験レポートの骨組みを例示している。あくまでも例示であって、全てをこの通りに従う必要はないが、指示された項目を含めた上で、報告書として他者が読みやすいレポートとなるよう考慮すること。

## グループメンバ

（補足：レベル毎に 全員が協力して実施 した上で、レベル毎にレポートをまとめる担当者を決め、全体を一つのレポートとして整理すること。分担方法も自由である。）

- 145763C 仲村 大地: Level1.1, 2.1
- 145738B 西銘 明留: Level1.2, 1.3
- 145717K 泉川真理南: Level2.2, 3.0
- 145714E 豊美 玲 : Level2.3

## 提出したレポート一式について

レポート一式は“shell:/net/home/teacher/tnal/2015-search1-mon/group2/” にアップロードした。提出したファイルのディレクトリ構成は以下の通りである。

（補足：必ず下記のように整理しろという指定ではない。自分たちでやりやすいように Level 毎に整理しても構わない）

```
./src/      # 作成したプログラム一式
./report/   # レポート関係ファイル、図ファイルを含む、
```

# 1 Leve l1: 最適化とは

## 1.1 Level 1.1: コンピュータと人間の違いを述べよ

### 1.1.1 課題説明

コンピュータが人間より得意とするモノ、その反対に人間より不得手のモノ、両者について2つ以上の視点(立場や観点など)を示し、考察する。

### 1.1.2 考察

- 視点 1: 計算

コンピュータならば、短時間で複雑な計算が可能であり、私たちも普段複雑な計算をする際にはコンピュータまたは計算機を用いていることから得意であることが明らかである。

だが人間は、簡単な計算ならば、計算しなくても暗算で求めることができるため、複雑な計算はコンピュータ、容易な計算は人間が得意である。

- 視点 2: 認識

人間は視覚、聴覚、感覚、嗅覚、味覚の五感があり、それを組み合わせて用いることで対象を正確に認識することができ、初めて見るものでも記憶にあるものと比較しそれがなんであるか推測することが可能である。

コンピュータは、様々なセンサなどで検出したデジタルデータを元に対象を認識しているため、センサで検出したデータが不足している場合、正しく認識することができない。

よって認識については人間のほうが得意であるが、Siri など音声を認識し、人間を会話できるほどの機能をもつアプリケーションも存在するので、今後は人間より認識できるようになる可能性がある。

- 視点 3: 学習・探索

コンピュータは、単純作業を繰り返し行う場合に短時間で行うことが可能であるため、パラメータを細かく変更して最適値を求めたり、それぞれのデータに重みをつけて分別またはクラスタリングしたりすることが得意である。人間は、「考えること」が可能である。認識でも話したが、対象を過去に五感で感じ、学習したものと比較し、推測することができたり、未知の物事について研究し、それがなんであるかを考え学習することができる。

- 視点 4: 記憶

コンピュータは、記憶を劣化せずに保持することが可能なため、検索が得意である。また、学習・探索で話したが、単純作業が得意なため、ソートも人間より早く正確に行うことができる。

人間は、記憶を劣化させてしまうため、検索・ソートについては不得意である。

- 視点 5: ゲーム

コンピュータは、ルールが単純であるゲームであれば、現状を評価し、最適な手段を探索することができる。現在ではチェスはコンピュータに勝つことができなくなっている。

人間は、ルールを理解し深めることでどんなゲームで最適な手段を探索でき、最適な手段が複数あった場合に状況にあった手段を選択することができる。

## 1.2 Level 1.2: 住宅価格を推定するモデルについて

### 1.2.1 課題説明

Housing Data Set[2] における RM (平均部屋数) から MEDV (平均価格) を推定するためのモデルについて検討した。

### 1.2.2 モデルへの入力

入力=RM(平均部屋数)

### 1.2.3 モデルにおける処理内容

散布図からおおよその傾向を読み取った結果，  
10\*RM-40 で MEDV を推定できるのではないかと考えた．

### 1.2.4 モデルの出力

出力=入力 RM から推定された MEDV(平均価格)  
出力結果を以下の図で示す．緑の線でかかれたモデルが推定された MEDV を表すものである．

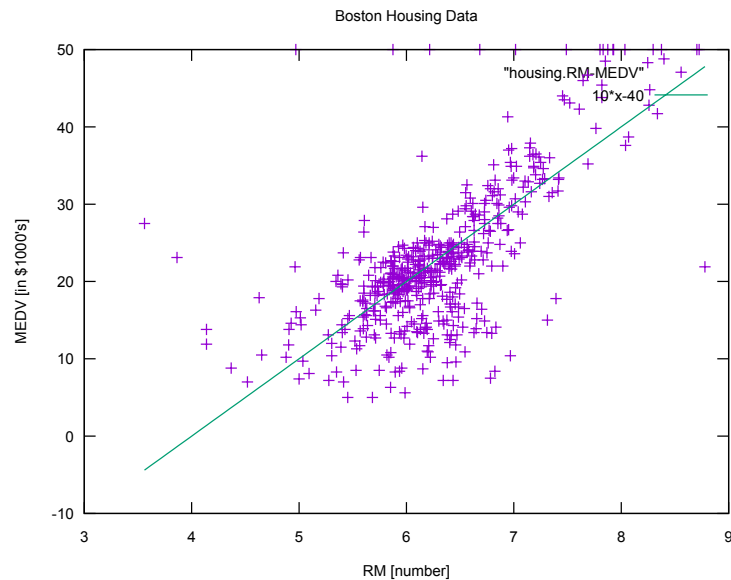


図 1: 探索初期値の変化

## 1.3 Level 1.3: モデルの良さを評価する方法について

### 1.3.1 課題説明

Level 1.2 で検討したモデルの適切さを評価する指標について検討した。

### 1.3.2 評価に用いる情報源

RM(平均部屋数),MEDV(平均価格),Level1.2 で推定された MEDV

### 1.3.3 評価手順

RM と MEDV から，おなじ RM の MEDV のデータから MEDV の平均値をとる．  
この平均値と推定された MEDV の差を比較する．

### 1.3.4 評価に基づいた適切さを計る方法

差が小さいほど適切なモデルである．

## 2 Level 2: 最急降下法による最適化

### 2.1 課題説明

3種類の連続関数  $y = x^2$ 、 $z = x^2 + y^2$ 、 $y = -x \times \cos(x)$  について、最急降下法の適用を通して探索挙動を観察した。以下ではまず共通部分である最急降下法の探索手続きについて、フローチャートを用いて解説する。その後、3種類の関数毎にプログラムの変更箇所、観察意図観察方法、観察結果、考察について説明する。

### 2.2 Level 2 共通部分

#### 2.2.1 探索の手続き (フローチャート)

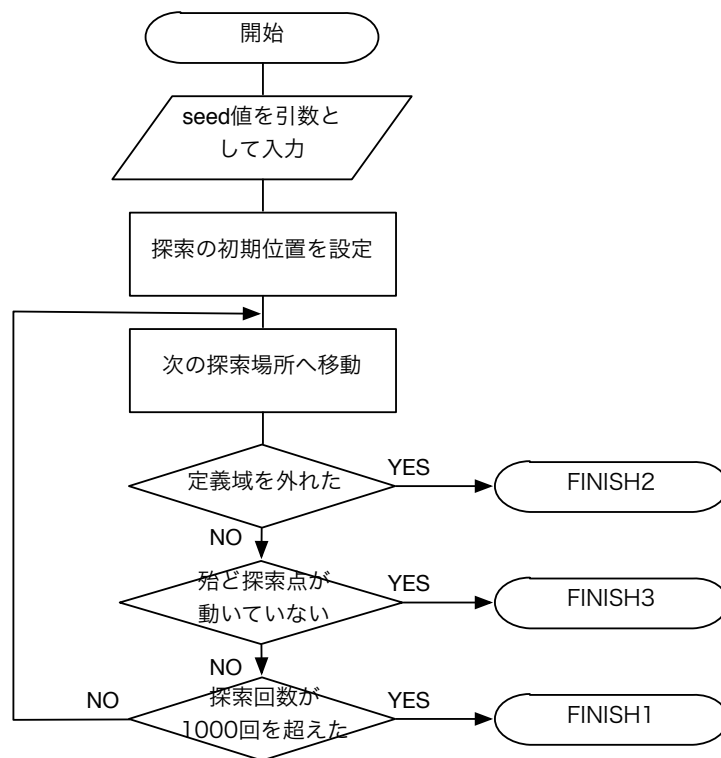


図 2: フローチャート

### 2.3 Level2.1: $y = x^2$ について

#### 2.3.1 プログラムソース (変更部分)

```
行数 変更点
8   #define Y_MIN 0.0    /* 定義域の最小値 */
35   z = x * x;
47   z_dx = 2*x;
```

### 2.3.2 観察意図と観察方法

seed と alpha の 2 つのパラメータがあるため, alpha の値を固定し, seed の値を変化させる実験と seed の値を固定し, alpha の値を変化させる実験の 2 つを行い, step 数, 探索の推移を観察する.

また, step 数を観察する際に run\_ave.sh を用いて行い, 探索の推移を観察をする際に trans\_step\_vs\_func.sh と trans\_x\_vs\_func.sh を用いて行った.

#### 1. alpha の値を固定し, seed の値を変化させる実験

観察した seed 値は, 1 ~ 10 までの小さい場合と, 1000, 2000, ..., 100000 の大きい場合の観察を行った. この際, alpha は 0.1 で固定している.

#### 2. seed の値を固定し, alpha の値を変化させる実験

観察して alpha 値は, 0.1 から 0.9 までの範囲の観察を行った, この時 seed 値は 1 で固定

### 2.3.3 実行結果

実験 1 では, step 数が少なくなっていくとある程度低くなるとまた徐々に高くなるという波のような変化している. 実験 2 では, step 数に大きな変化は起きなかったが探索の推移を観察より, 探索の初期位置が大きく変わっていることがわかった. また, 変化が alpha と類似していた (図 6)

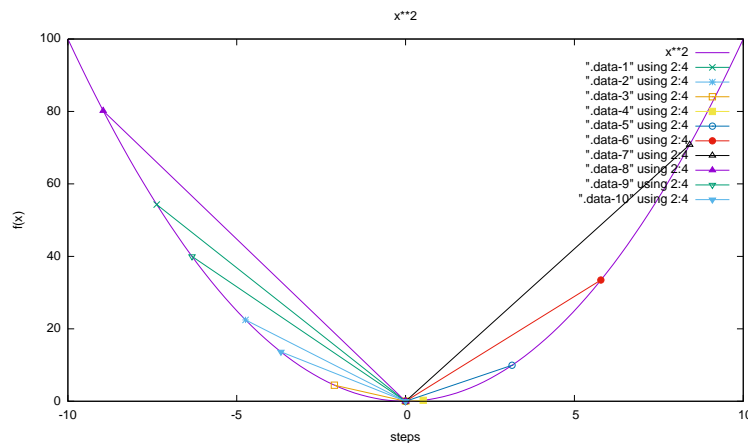


図 3: 探索初期値の変化

この図は, seed 値を 1 ~ 10 まで変化させ, それぞれの初期探索位置をグラフ化したものである. このとき, 初期探索位置がわかりやすいように, 探索回数の少ない alpha を使用している.

### 2.3.4 考察

結果より, alpha の値を変動させることによって, step 数を大きく変動させることができることがわかった. 効率を考えると, 探索回数が 1000 回と限られているため, できるだけ早く最小値を求めることが望ましいと考える. だが, 解の質を考えた場合, すぐに 0 に収束してしまうため情報として最小値だけになってしまい情報量が少ない. step 数を多くすると探索の推移が細かに把握できるため, その分情報量が大きくなり, 解の質としては向上すると考えた.

## 2.4 Level2.2: $z = x^2 + y^2$ について

### 2.4.1 プログラムソース (変更部分)

行数 変更点

```
32  z=x*x+y*y;  
44  z_dx = 2*x*z_dx;  
56  z_dy = 2*y*z_dy;
```

### 2.4.2 観察意図と観察方法

学習係数が探索挙動に及ぼす影響を確認するために、seed 値と alpha 値を変更し観察を行う。

1. alpha 値を固定して seed 値を変更する。  
alpha 値は 0.1 で固定、seed 値は 1 10,1000 刻みの 1000 10000 とする。
2. seed 値を固定して alpha 値を変更する。  
seed 値は 1 で固定、alpha 値は 0.1 0.9 とする。

### 2.4.3 実行結果

1.seed 値を変更したときの step 数。  
step 数に多少の変化がみられたものの、劇的な変化ではなかった。

2.alpha 値を変更したときの step 数。

alpha 値を変更したことにより、step 数に大きな変化がみられた。  
特に alpha 値を 0.1 から 0.2 に変化させたときに step 数の大幅な減少がみられる。

表 1: alpha 値を変更したときの step 数の変化

alpha 値	step 数
0.1	686
0.2	184
0.3	82
0.4	45
0.5	26
0.6	16
0.7	19
0.8	16
0.9	40

また step 数が変化すると探索ステップ数あたりの目的関数推移図にも変化が生じることがわかった。

alpha 値 0.1(step 数が多い) ときのほうが図がなめらかである。

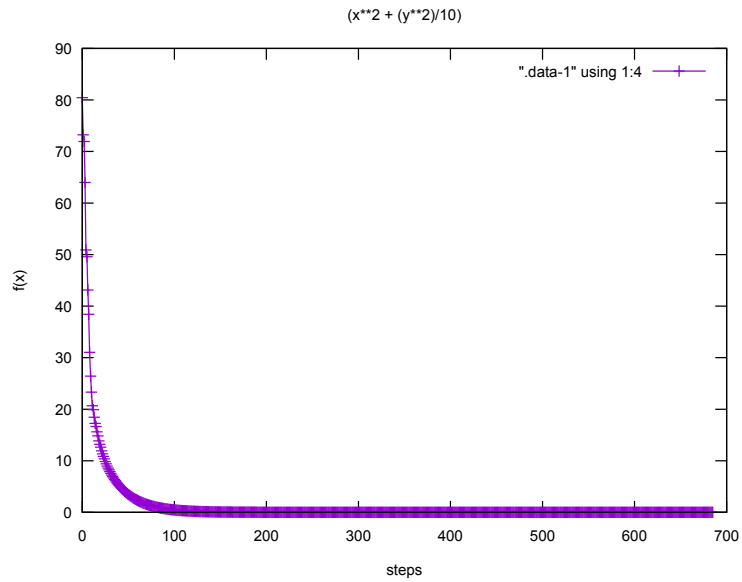


図 4: alpha 値 0.1 のときの推移図

#### 2.4.4 考察

学習係数を変化させることにより step 数を減らすことができた。探索回数が減ることにより効率はあがったと考える。しかし、探索ステップ数あたりの目的関数推移図を見ると、step 数が多いほうが探索点が最適解にたどりついたより正確な地点を知ることができると思う。

## 2.5 Level2.3: $y = -x * \cos(x)$ について

### 2.5.1 プログラムソース (変更部分)

行数 変更点

```
28  z = -x * cos(x);
36  z_dx = x * sin(x) - cos(x);
```

```
#!/bin/bash
set -e

# 図タイトル(func), シード値(seed) の取得。
if [ $# -eq 3 ] ; then
seed1=$1
seed2=$2
seed3=$3
else
echo "Usage: prompt> $0 \"gnuplot_style_func\" seed"
exit 0
fi

# シミュレーション実行&データ抽出。
```

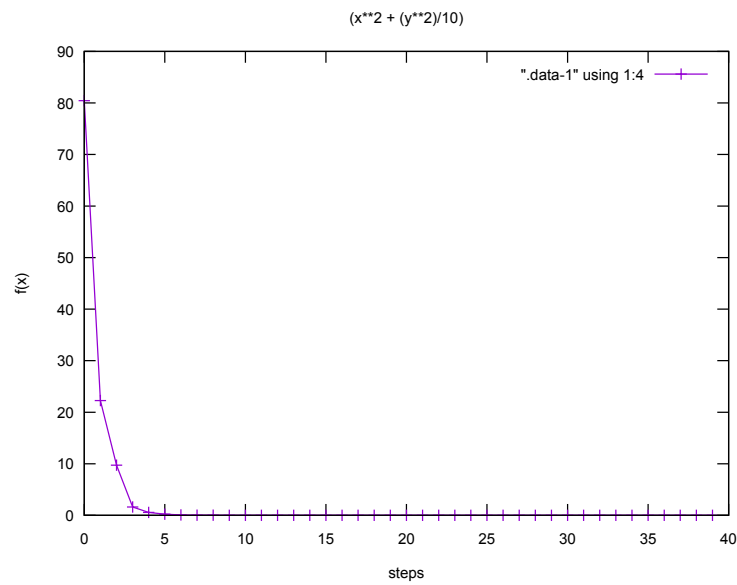


図 5: alpha 値 0.9 のときの推移図

```
exec_file="./steepest_decent"
transition_file="./transition.txt"
exec_file1="./Compare"

if [ -f $transition_file ] ; then
    rm $transition_file
fi

for seed in $seed1 $seed2 $seed3
do
    archive_file=.archive-$seed
    $exec_file $seed > $archive_file
    cat $archive_file | cut -f2,4,6,8 -d" " > .data-$seed
done

echo "seed $seed1 finish f(x,y)"
tail -c 14 .data-$seed1 | cut -c-13
seed1min='tail -c 14 .data-$seed1 | cut -c-13 '

echo "seed $seed2 finish f(x,y)"
tail -c 14 .data-$seed2 | cut -c-13
seed2min='tail -c 14 .data-$seed2 | cut -c-13'

echo "seed $seed3 finish f(x,y)"
tail -c 14 .data-$seed3 | cut -c-13
seed3min='tail -c 14 .data-$seed3 | cut -c-13'
```



```
echo "最適解は '$exec_file1 $seed1min $seed2min $seed3min' である"
--
以下，作図
```

### 2.5.2 観察意図と観察方法

まずは，元のプログラムを必要な部分だけ変更し，seed 値と alpha の値を変更して実験する．

また，この数式は局所解が複数あり，最適解を探すことが困難である．よって，この問題を解決するために，初期位置の設定を複数のシード値を用いて位置の変更を行い，複数の局所解の大きさを比べ，より小さい方を最適解とするプログラムを作成した．

#### ・観察方法

1. alpha の値を固定し，seed 値を 1-10 までの小さい値と，1000-10000 までの 1000 刻みの値とで観察していく．
2. 異なる探索点 (seed 値) から複数回プログラムを実行し，局所解を比較， $f(x,y)$  の一番小さい値を最適解として出力．

### 2.5.3 実行結果

1. alpha の値を固定し，seed 値を 1-10 までの小さい値と，1000-10000 までの 1000 刻みの値とで観察していく．

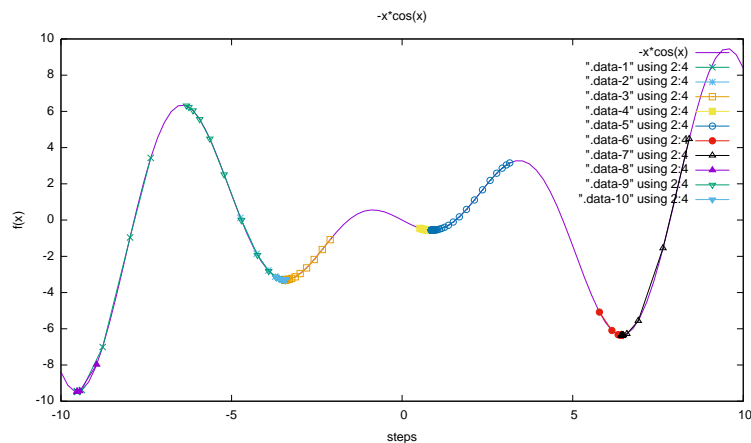


図 6: 探索初期値の変化

1000-10000 までも，同様の結果となった．

2. 異なる探索点 (seed 値) から複数回プログラムを実行し，局所解を比較， $f(x,y)$  の一番小さい値を最適解として出力．

```
% sh trans_x_vs_func_seed.sh 1000 2000 3000
FINISH 3 step 54 x and y were not updated.
FINISH 3 step 7 x and y were not updated.
FINISH 3 step 65 x and y were not updated.
seed 1000 finish f(x,y)
-0.5610963382
seed 2000 finish f(x,y)
-9.4772942595
seed 3000 finish f(x,y)
```

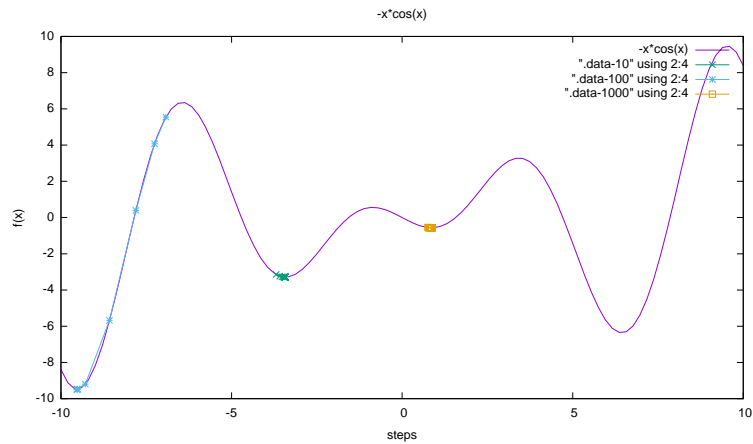


図 7: 探索初期値の変化

-0.5610963382

最適解は -9.477294 である

#### 2.5.4 考察

1. まず, seed 値を変更し, 単体で実行したプログラムを考察していく.

初期位置が一定数値を超えなければ, 定義域内での最適解に到達できず, 他の局所解を最小値として出力する.

このプログラムでは, 最急降下法を用いているため, 切片の傾きが 0 となる場所を目指していく. しかし, この数式だと切片の傾きが 0 となる最適解でない局所解を算出してしまう結果となった.

これを解決するために, 2 つ目の方法を使用した. 2. 次に, 複数の seed 値を用いて比較したプログラムを考察していく.

比較には C 言語で作成したプログラムを用いた. 先述した seed 値を変更して同じ図に出力するシェルプログラムから最終探索値を引数として取り, その値を比較して出力した. このプログラムでは, 3 つの異なる seed 値を使用する事によって, 異なる初期位置から探索をする事ができる.

それぞれの初期位置からそれぞれの局所解を探し出す事で, 解の質は大きく向上したと考える.

また, 学習係数 alpha の値を変更して探索を行うと, 極端に探索回数が少なくなったが, 稀に極端に探索回数が大きくなるという事が発生した. 今回試した中での alpha の最適値は 0.5 と判断した.

### 3 Level 3: 最急降下法が苦手とする状況

#### 3.1 課題説明

最急降下法が苦手とする状況についてその理由を解説し、検討した改善方法について解説する。

##### 3.1.1 原因

「山 (谷)」の数は一つにも関わらず最も勾配の高い方向に移動するため、一直線に谷には向かわずジグザグ移動してしまい、効率的ではない。

##### 3.1.2 改善方法

- ・複数の初期値から探索を行いより短い探索で解にたどり着ける初期値を探す。

最適解の近傍に初期値を設定することができれば、余計な探索を減らすことができると考えた。最適解の  $x$  軸方向の直線上に近い場所に初期値を設定することができれば、一直線に谷に向かうことも可能となる。

- ・また収束速度を安定させる手法として、前回得られた勾配ベクトルと今回得られた勾配ベクトルの符号の変化によって刻み幅の調整をする方法がある。符号が反転したときに刻み幅を小さくし、符号が等しいときに刻み幅を大きくすることで安定して収束する。

(補足：参考文献は thebibliography 環境を使って列挙し、本文中で適切な箇所引用するようにしましょう。例えば下記文献は、アブストラクトや Level 4 で引用しています)

### 参考文献

[1] 情報工学実験 2: 探索アルゴリズムその 1 ( 當間 )

<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/2015/info2/search1/>

[2] Housing Data Set

<http://archive.ics.uci.edu/ml/datasets/Housing>