

# 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 0

2014 年 12 月 14 日

## 概要

この骨組み（テンプレート）を利用するには、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。）

- 945734J 当間愛晃: 担当 Level1.1, 1.2
- 945700 hoge: 担当 Level2.1, 2.2
- 945700 hoge: 担当 Level2.3, 3.1
- 945700 hoge: 担当 Level3.2

## 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/2013-search1-mon/group0/” 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: 探索とは

## 1.1 Level1.1: 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。

- ・コンピュータはパターン化されたものを決まった動作を繰り返す。
- ・コンピュータは大量の情報を取り扱いや特定のパターンに基づいてまとめたりする。
- ・コンピュータの不得意なことは、感情移入判断することが出来ない、決まったことしか出来ない。
- ・コンピュータと違いは、パターン化されていないことも出来る

### 1.1.2 考察

- 視点 1: hoge  
コンピュータならば\* \*が可能であり云々  
コンピュータは、パターン化されたことが得意
- 視点 2: fuga  
人間は\* \*しなくてはならないため云々  
人間は、パターン化されないことでも出来る

## 1.2 Level1.2: 評価方法（目的関数の設計指針や方法）について

### 1.2.1 課題説明

Amazon における書籍検索時に「ファンタジー作品で泣ける作品」を探し出すためのアイテム集合  $x$  と目的関数  $f(x)$  について検討した。

### 1.2.2 アイテム集合 $x$ について

我々のグループでは、ファンタジー作品のレビューに書いてあるワードをアイテム集合  $x$  とすることを考えた。次のようなレビューを例にとると、レビューに書いてあるすべての単語がアイテム集合  $x$  になる。

いやあ、おもしろい。異世界ファンタジーと言うのは、最初の数ページでこれはちょっと、と思う場合と、そこで引き込まれてあっという間と言う場合の二種類がある気がするけど、本書はもちろん後者の方。ありえなあいと叫ぶだけのような、そんな浅いファンタジーではなく、とてもとても重厚な、しっかり細部も練り込まれた、実に味わい深い作品でした。

### 1.2.3 目的関数について

目的関数  $f(x)$  はすべてのアイテム集合  $x$  (レビュー内容の単語) に対して重み付けをし、それを足し合わせた合計を返す関数とする。先ほどのレビューを例にとると、「おもしろい」や「重厚な」、「味わい深い」などの作品を高く評価する単語に対しては他のワードより大きな値を付加する。そして探索目的は値が最大になるような  $f(x)$  である (図 1 参照)。

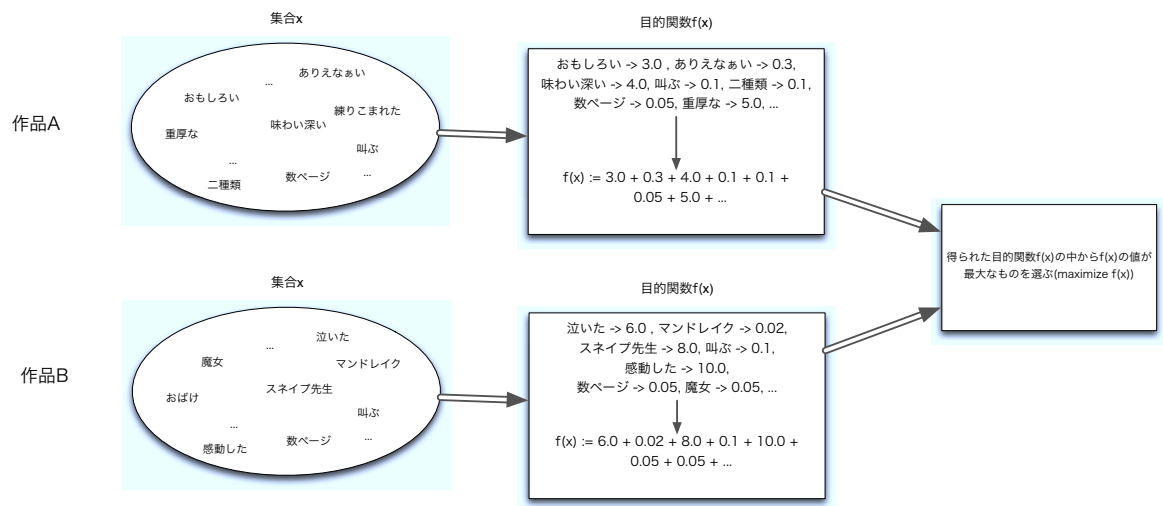


図 1: 泣ける作品を探し出すまでの流れ

#### 1.2.4 目的関数の欠点

この目的関数の設計上の問題点はあらかじめワードとワードに対応する重みの値を定義しておく必要があるということである。

## 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  $y = x^2$ 、 $z = x^2 + y^2$ 、 $y = -x \times \sin(x)$  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

( 補足 : Level2.1, 2.2, 2.3 には共通する部分が多いため、共通部分は独立して報告すると良いでしょう )

#### 2.2.1 探索の手続きとフローチャート ( 共通部分 )

( 手続きとフローチャートはまとめて一つの節にしても構いません )

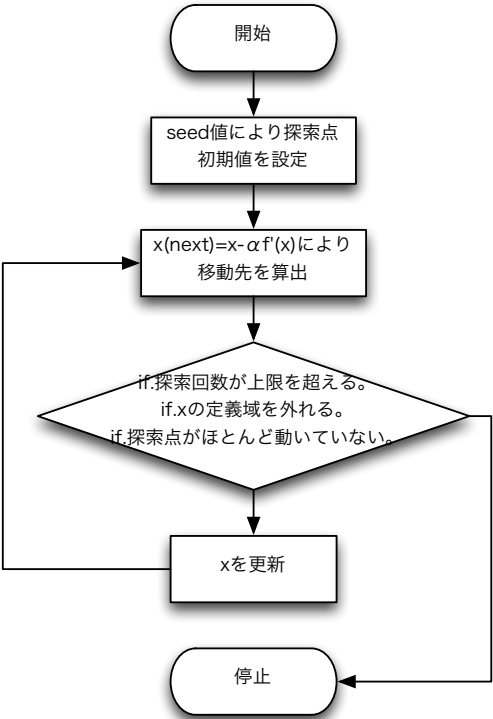


図 2: 探索手続きのフローチャート

### 2.3 Level2.1: $y = x^2$

#### 2.3.1 プログラムソース

ソースコード 1: 変更された関数  $f(x, y)$

```
1 double f(double x, double y) {
2     double z;
3
4     /** 以下の式を編集して完成させよ(1) **/
5     z = x*x;
6 }
```

```

7   return( z );
8   }

```

## ソースコード 2: 変更された $f(x,y)$ の微分値

```

1 double pd_x(double x, double y) {
2     double z_dx;
3
4     /** 以下の式を編集して完成させよ(2-1) **/
5     z_dx = 2*x;
6
7     return( z_dx );
8 }

```

### 2.3.2 観察意図と観察方法 (実験計画)

変更したプログラムに対して、シード値を 1、 $\alpha$  値を 0.1 として実行する。run\_ave.sh ファイルの実行結果により、シード値が及ぼす挙動についてみれるはずなので、その結果をもとにシード値の考察を行う。次に  $\alpha$  値により探索点の推移距離が変わるので、シード値を固定したまま  $\alpha$  値をより大きく、または小さくした値に変更して実験を行うことで  $\alpha$  値が探索挙動に及ぼす影響を考察する。この時、関数に対して谷が 1 つなので  $x$  の解や  $\alpha$  値が 1 つに定まるはずなのでそれを意識する。

### 2.3.3 実行結果

#### ソースコード 3: 'steepest\_decent 1' の実行結果

```

1 step 0 x -7.3692442371 y 5.1121064439 f(x,y) 54.3057606266 5.430576e+01
2 step 1 x -5.8953953897 y 5.1121064439 f(x,y) 34.7556868010 3.475569e+01
3 .
4 .
5 .
6 step 62 x -0.0000072277 y 5.1121064439 f(x,y) 0.0000000001 5.224013e-11
7 step 63 x -0.0000057822 y 5.1121064439 f(x,y) 0.0000000000 3.343369e-11
8 .
9 .
10 .
11 step 74 x -0.0000004967 y 5.1121064439 f(x,y) 0.0000000000 2.466971e-13
12 FINISH 3 step 75 x and y were not updated.

```

シード値を 1、 $\alpha$  値を 0.1 に設定して実行した結果、 $x$  の値が -7.3692442371 から始まり、step63 で  $f(x)$  の値が 0、探索点がほとんど動かなくなった為 step75 で終了した。

./run\_ave.sh を用いてシード値を千刻み 10 パターン (1000 ~ 10000) で実行し平均試行回数を計測した。1000 ~ 10000 までそれぞれの試行回数は順に 64、75、69、74、71、73、73、72、74、70 であり、平均試行回数は 71.5step であった。

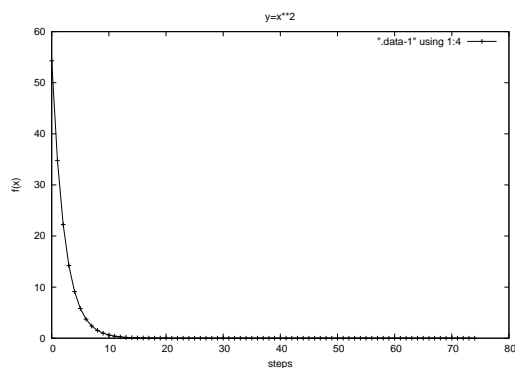


図 1 step あたりの目的関数推移線グラフ

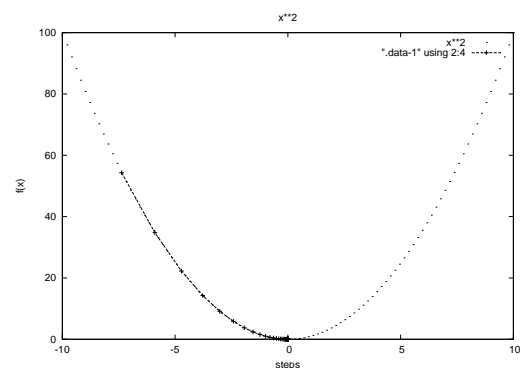


図 2 探索点が関数上をどのように推移したかを示すグラフ

図 1 と図 2 はシード値 1、 $\alpha$  を 0.1 に設定して実行された最急降下法アルゴリズムにおける点の step あたりの移線グラフと関数上での推移グラフである。

### 2.3.4 考察

シード値は探索点の初期値の設定に用いるものである。実験結果よりシード値による高さの決定はランダムであるが、値を固定した場合は、決定した高さがより低いほうがstep数は少ない事が分かった。さらにコマンド「tail -1 .archive-\*」を実行した結果、高さと得られる解の質は関係がないと考えられた。

値に関する考察を得るためシード値 1、 値 0.1 における実験に対して 値を変更した 2 つの実験を行う。シード値を固定したまま 値を 0.3、0.007 に変更する。シード値は固定であるため開始地点は同じになる。

ソースコード 4: 値 0.3、0.007 に置ける 'steepest\_decent 1' の実行結果

```
1  値
2  0.3
3  step 20 x -0.00000000810 y 5.1121064439 f(x,y) 0.0000000000 6.565164e-15
4  FINISH 3 step 21 x and y were not updated.
5
6  値 0.007
7  step 983 x -0.0000070539 y 5.1121064439 f(x,y) 0.0000000000 4.975774e-11
8  FINISH 3 step 984 x and y were not updated.
```

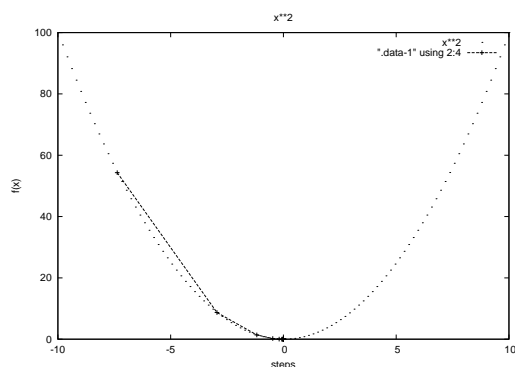


図 3 探索点の関数上での推移を示すグラフ ( 値 0.3)

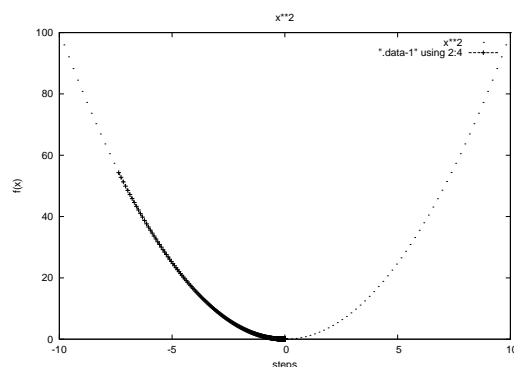


図 4 探索点の関数上での推移を示すグラフ ( 値 0.007)

値を変更する事で推移距離が変わる。 値を 0.3 にすると推移距離が長くなり、小さくすると短くなる。

値を 0.3 にすると、 値 0.007、0.1 に比べ、step21 で探索が済んでいるため効率性がよく、得られる解の質に関しても、誤差が小さくなっていることが実行結果より判断出来る。

値を小さくすると誤差がより大きくなってしまったことに関しては、小さいと探索点から動いていないと判断されてしまい、探索を打ち切られてしまうからと考えられる。

これらの実行結果より、3 つの 値の内 0.3 付近がより、探索における有用性が見いだせる。今回の場合、最小値の値が 0 であり、傾きの値が 0 となる点であることが自明である。よって移動先の算出は式から見いだせ、その値は 値が 0.5 のときである。つまり 値が 0.5 の時 step2 で最適値を見つけ出す事ができ、解の質においても誤差はない。今回の実験で、0.3 付近でより有用性が見いだせたのは、より 0.5 に近かったからと判断できる。

## 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 )

ソース変更 :

```
/** 以下の式を編集して完成させよ (1) **/  
z = x * x + y * y;  
/** 以下の式を編集して完成させよ (2-1) **/  
z_d x = 2 * x;  
/** 以下の式を編集して完成させよ (2-2) **/  
z_d y = 2 * y;  
この三カ所である。
```

### 2.4.2 観察意図と観察方法

alpha の数字を変えてみて、どのようになるのかを確認して行く。

`/transxyvsfunc.sh''x**2+y**2''1` を用いて、グラフの図を作成しどのようになっているのかを確信した。

### 2.4.3 実行結果

alpha = 0.1 の時

`/Users/e135711/実験 2/steepestsearch./transxyvsfunc.sh''x**2+y**2''1`

FINISH 3 step 75 x and y were not updated.

alpha = 1.0 の時

`/Users/e135711/実験 2/steepestsearch./transxyvsfunc.sh''x**2+y**2''1`

FINISH 1 step 1000 this trial couldn't be searchenough under the term  $\epsilon = 1000$ .

### 2.4.4 考察

x 軸と y 軸から見て“ .data-1 ” using 2:3:4 を確認すると、alpha の数を変えてみた結果、数字を上げて行くと計算回数が少なくなり、数字を下げると計算回数が多くなった。

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

### 2.5.1 プログラムソース (変更部分)

### 2.5.2 観察意図と観察方法

### 2.5.3 実行結果

### 2.5.4 考察

### 3 Level 3: 最急降下法が苦手とする状況

#### 3.1 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

傾きが小さくなるほど移動距離が小さくなって検索点が多くなっている。谷に直線的に向かわない。

##### 3.1.2 改善方法

$\alpha$  を大きくとり、移動距離を大きくする。今居る場所より次に移動する場所を移動する前に比較して高かったら、 $\alpha$  の値を下げもう一度比較していく。



## 4 Level 4: モデル推定時における目的関数の設計

Housing Data Set[2] を例に、モデルの適切さを図るための目的関数に付いて設計した。

### 4.1 目的関数について

設計した目的関数は最小化である。点と線の最短距離の和が小さければ小さいほど適切。点と線の横軸と縦軸の距離の和が小さければ小さいほど適切。

### 4.2 設計理由について

モデルと全ての点の差を出して差が小さい方がモデルとして適切なので、モデルとの差が分かるような目的関数にした。

（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で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/2013/info2/search1/>

[2] Housing Data Set

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