3章　関数の最適化

- **最適化** ... 与えられた制約条件のもとで関数の値の最大(最小)となる変数を求めること。

勾配法

関数の最適化方法として基礎的で単純だが強力。ニュートン法などの基となるのでここから紹介する。

基本的には、**与えられる関数 f'(x) = 0 は制約の中で f'(x)=0 となる x は1つしかない**ことが条件となる。勾配の方向に進んで行くのが最も関数の値を変化させるため、勾配の方向に1回で移動する幅(ステップ幅)を設定し、繰り返し更新を行っていく。また、関数f(x)とその導関数f'(x)が既知である必要がある。

1変数の場合

具体的な手順は以下の通りである。

---

algorithm 1. １変数勾配法

---

1. xの初期値x0を与え、ステップ幅hをh0とする。
2. x, hを次のように更新する。

$$h \leftarrow sgn(f'(x))|h|, \quad X \leftarrow x, \quad X' \leftarrow x + h$$

3. もしf(X) < f(X')であれば次の計算を行う。
    - f(X) ≧ f(X')となるまで次の計算を行う。

$$h \leftarrow 2h, \quad X \leftarrow X', \quad X' \leftarrow X + h$$

    - f(X) < f(X')となったら次を行う。

$$x \leftarrow X, \quad h \leftarrow \frac{h}{2}$$

4. そうでなければ次の計算を行う。
    - f(X) ≦ f(X')となるまで次を行う。

$$h \leftarrow \frac{h}{2}, \quad X' \leftarrow X' \, h$$

- f(X) > f(X')となったら次を行う。

$$x \leftarrow X, \quad h \leftarrow 2h$$

5. ステップ2に戻り、これを |f'(x)| ≦ |ε| になるまで繰り返す。(εは許容誤差)

6. 以上より得られたxを返す。

勾配法はコンピュータで行う都合上、解析的に解いた場合と比べて誤差がある程度出てしまうのは仕方がないので、ステップ5で行うような収束判定が必要である。また、初期値の与え方次第で収束にかかる時間がかなり変わるらしいので、適切な初期値の与え方、すなわち

多変数の場合

多変数の場合も1変数と変わらず勾配∇f(x)が最も関数値を変化させるので、勾配∇fを与えてその直線方向で最大値になる点を探索する(直線探索)。探索の方法は1変数の場合と変わらないが、アルゴリズムを以下に示す。

---

algorithm 2. 多変数勾配法

---

1. **x** に初期値を与える。
2. 関数F(**x**) = f(**x** + t∇f(**x**))に対してalgorithm 1 を行い、直線探索を行う。
3. algorithm 1に返される t を用いて次の更新を行う。

$$\Delta \mathbf{x} \leftarrow t \nabla f(\mathbf{x}), \quad \mathbf{x} \leftarrow \mathbf{x} + \Delta \mathbf{x}$$

4. ステップ2に戻り、これを||Δ**x**|| < δ となるまで繰り返す。
5. 得られた**x** を返す。

ステップ2で必要となるのが、F(**x**)の導関数F'(**x**)なので、これを求める。**x** = **x0** + t ∇f0 とおいてF(t) = f(**x**(t)) をt で微分すると次のようになる。(自分用メモ：**x**がtの関数になっている)

$$\frac{dF}{dt} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \frac{\partial f_0}{\partial x_i} = \nabla f \cdot \nabla f_0$$

このF(t)が極値を取る、すなわち

$$\frac{dF}{dt} = \nabla f \cdot \nabla f_0 = 0$$

となるので、探索直線∇f0 と極値を取る点**x** での勾配∇f は直行することがわかる。 これは、初期値に基づく探索直線(1)で得られた値を再び初期値として探索直線(2)がある時、これらは直行するということを示している。

このこと自体に意味はないが、関数を等高線表示した場合に勾配法がどのように更新しているのかがわかりやすくなるので嬉しい。

勾配法の問題点

斯様にして万能にも思える勾配法だが、以下のような欠点がある。

1. アルゴリズム内に必ず勾配∇f が必要になるため、不連続であったり解析的に勾配が計算できない場合は適用できない。
2. 局所解から抜ける気がないので、関数の形次第で初期値の与え方で収束する値が異なる可能性がある。
3. 極値付近の形状次第で収束に時間がかかることがある。

ニュートン法

f(**x**)について2階の導関数まで求められる場合、より効率的な**ニュートン法**を適用することができる。多分世の中の人はコンピュータで関数近似をする際に最初に勉強するのはこれ。f'(x) = 0となるxが最大になるのでうれしいよねということを考える。

1変数の場合

1変数関数f(x)をテイラ展開するとこんな感じになる。

$$f(\bar{x}+\Delta x) = f(\bar{x}) + f'(\bar{x})\Delta x + \frac{1}{2}f''(\bar{x})\Delta x^2 + \dots$$

ここで右辺の3次以上の高次の項を無視して放物線として考える。左辺が最大になるΔxを求めるならば、次を考える。

$$f'(\bar{x}) + f''(\bar{x})\Delta x = 0$$

$$\Delta x = -\frac{f'(\bar{x})}{f''(\bar{x})}$$

このΔxを用いて、最適解xのより良い近似値は次のようになる。

$$x = \bar{x} - \frac{f'(\bar{x})}{f''(\bar{x})}$$

これを反復していくことで最適解x* を求める。以下にアルゴリズムをまとめる。

---

algorithm 3. 1変数ニュートン法

---

1. 初期値xを与える。
2. 次のようにxを更新する。

$$\bar{x} \leftarrow x$$

$$x \leftarrow \bar{x} - \frac{f'(\bar{x})}{f''(\bar{x})}$$

3. ステップ2に戻り、これを | x - x~ | < δ となるまで繰り返す。
4. 得られたxを返す。

共役勾配法