計算天文学 II 第 8 回 最適化 (1)

牧野淳一郎

1 最適化

今日と来週で最適化の話をする。

これは天文学の観測・理論のあらゆる場面で必要になるとても重要な技術である。

現代の天文学の観測においては、例えばなにかを観測してデータをとったとして、それから実際に天文学的に意味があることをいうまでにはいろいろなステップがはいるのが普通である。多くの場合に、これはいろいろな自由パラメータがあるモデルを持ってきて、そのモデルのパラメータを観測データを「もっともうまく説明する」ように決めるということである。例えば銀河内のガスの速度から質量分布を推測するとか、銀河団ガスからの X 線放射から質量を推定するといった場合には、結局そういうことをやっているわけである。もっともうまく説明するとは、具体的にはなんらかの形で誤差を表現して、それを最小化するということである。

これは、まあ、形式的には例えばこういうふうな話になる：

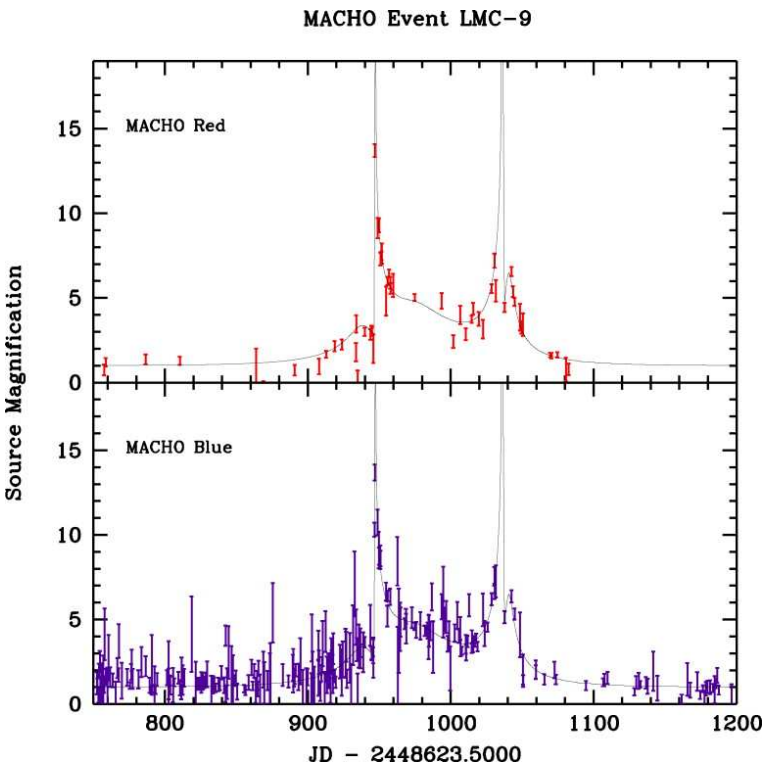
「ある領域 D 上で定義された実数値関数 $f(x)$ がある。その最小値とそれを与える $x \in D$ を求めよ」

つまり、最適化というのは要するにこういう話である。

とはいえ、実際にどうやって上の問題の答を求めるかというのは、もちろん領域 D がどんなものかと関数 f がどんなものかによる。例えば、観測データを線形回帰して直線近似するなら、D は直線 $y = ax + b$ の係数 (a, b) の集合ということになろう。 f は 2 乗残差である。これは 2 次形式の最小化になり、微分すれば連立一次方程式が出てきて解ける。パラメータの数が多くても、2 次形式なら話は同じである。

これに対して、同じような多次元空間内の最適化でも、もとの関数がどんなものか良くわからないとか、計算が面倒であるとかいうと、急に話がややこしくなる。

例えば、次の図は連星重力レンズと推定されたものの観測結果と、それのモデルである。



横軸は時間、縦軸は明るさである。連星レンズの特徴は、単一星の場合と違ってピークが 2 つできることと、そのピークが単一星のばあいよりもずっと明るいことである。

連星レンズの場合、パラメータの数は非常に多い。連星自体の軌道要素が 6 個、その他に質量比、周期、光源の速度、それぞれの我々からの距離ということで 10 個以上ある。なお、このうちいくつかは縮退しているので、本当のパラメータは 9 個である。で、どのパラメータを変えるとなにがどう変わるかというのは簡単にはわからない。こういう時に、どうやってレンズのライトカーブの観測から、物理的な意味を引き出せるのだろうか？

というわけで、世の中には多様な最適化手法がある。これらを簡単にまとめるのが今日の話ということになる。

2　最適化手法の分類

ここでは、本来分類されるべきものは手法ではなく、問題の分類があってそれに対応して手法があるのかもしれないが、まあ、実際には、問題の定義自体が、「こういう方法で解ける」というのをかっこよくいっただけというところもある。とりあえず、ここで考える分類は以下のようなものということにしておく。

- 決定論的方法
  - 連続関数向け方法
  - 離散的関数向け方法
- 確率的方法

決定論的方法とは、文字通り問題とやり方を決めればあとは機械的に計算が進んで、いつでも同じ答がでるものである。これに対して、確率的方法とは、良さそうな答を捜す時に乱数等を使ってある意味「適当」にやるものである。

適当にやるよりも、真面目にやったほうがいいに決まってるのではないかと思うかも知れないが、必ずしもそういう問題ばかりではない。例えば、ある種の問題では、まともにやって正しい答を求めるのに必要な手間が、問題の大きさの多項式よりも速く増大する。

この一例が巡回セールスマン問題というものである。これは、名前の由来は、「あるセールスマンが一日に沢山のお客を回る時に、どの順番で回るのがもっとも効率的か？」ということであるが、例えば電話線やネットワークの線をどう引くのか効率的かといった問題にも応用される、実用上はいろんなところで出てくる問題である。

素朴な解法は、全部の順序を調べてもっとも短いのを捜すというものだが、これは手間が回る場所の数 n の階乗で増えるので n が 10 を超える当たりから実際的ではなくなる。が、例えば n^p といった、n の冪乗の手間で正しい解が見つかるような方法は知られていない。

ところが、このような問題に対して「シミュレーテッドアニーリング」や「遺伝的アルゴリズム」といった、確率的な方法を使うと、それが本当にもっとも良い解であるという保証はないが、まあまあそんなには悪くない解が n^2 程度の手間で求まる。

まあ、この辺の詳しい話は来週にして、決定論的方法のほうの分類だが、連続関数というのは定義域が N 次元ユークリッド空間の連続な部分集合で、最適化したい目的関数も連続な実数値関数であるようなものである。そうでないものというのは要するにそれ以外の全部である。上の巡回セールスマン問題は後者の一例である。

この講義では、離散的な場合の決定論的方法はやらない。これは、問題によってあまりに沢山いろんな方法があるので、何をやるべきか良くわからないからである。

3　連続関数の最適化

ここで述べる手法は、とりえあえず定義域があまり変な形をしていなくて (というのをちゃんと定義することはできるが、やると話が長くなるので省略)、関数はその定義域のなかで連続で極小値を 1 つだけ持つという場合のためのものである。

この場合、目的関数 f が 2 階微分を持てば、原理的には話は極めて簡単になって、$\nabla f = 0$ という方程式をニュートン法で解けばいい。が、多くの場合にこれはあんまりうまくいかない。というのは、変数の数 n が多くなると計算しないといけない 2 階微分の数は n^2 に比例して増えるわけで、計算量が増えるだけでなく書かないといけないプログラムの量が増えるという問題があるからである。

微分を数値的にやればいいのではとも考えられるが、これも丸め誤差等の影響があって難しい場合が多い。

もっとも、微分を数値的にやるのではなく、数式的にやる、つまり、もとの関数の数式から数式処理プログラムに生成させたり、あるいはプログラム自体を「微分」する、つまり、目的関数を計算するプログラムから微分を計算するプログラムを自動生成するといった研究もかなり進んではいる。実際に問題を解こうという時には、こういった手法が使えないかどうかも考えてはみるべきであろう。

というわけで、以下はニュートン法とかではない方法。まず 1 変数、それから多変数にいく。

3.1 1 変数の場合

まあ、1 変数ならニュートン法でいいのではないかとも思うわけだが、多変数の場合のための準備ということで一応説明しておく。実際には、多変数の問題を解く時に形式的には 1 変数の問題の繰り返しになって、そこでは簡単には 2 階微分が計算できなくてニュートン法というわけにはいかないので、それ以外の方法がいるわけである。

良く本に載っているのは、黄金分割法というものである。これは、以下のような方法である。区間 $[a, b]$ のなかに関数 $f(x)$ の最小値があることはわかっているとする。

1. x_1, x_2 を、$[a, b]$ をそれぞれ $\sqrt{5} - 1 : 2$ およびその逆に内分する点とする。
2. それぞれの点で関数値 f_1, f_2 を計算する。
3. $f_1 > f_2$ なら最小値は $[x_1, b]$ にあるので、 a を x_1 で置き換え、1 に戻る。細かいことをいえば x_2 が 次の x_1 になるので、使い回せる。
4. そうでなければ逆に b を x_2 で置き換え、同様に 1 に戻る。
5. 上の全体を $|a - b|$ が十分小さくなるまで繰り返す。

なお、一般には別に黄金分割でなくても、区間内に適当に 2 点とってその大きい方を新しい区間の端にするというので構わない。黄金分割のミソは上の説明の「細かいこと」、つまり、分割点の一方を使い回せるので反復一度について関数計算が 1 度ですむということである。

一度ですむのはいいが、収束は遅い。一度反復した時に区間の幅が 0.62 倍 にしかならないからである。これはつまり一次収束で、反復毎に定数分の 1 になるものである。

もうちょっと賢い方法としては、疑似ニュートン法的なものがある。要するに 3 点あれば 2 次関数で近似できるので、その極値を求めようというものである。最適化問題ではない非線形方程式では Secant 法と呼ばれているものの拡張である。 Secant 法程速くはないが、 一次よりも速い収束をすることが知られている。

3.2 多変数の場合

多変数の場合にも、黄金分割にあたるような直接探索法というのはあることはあるが、あんまり使えないので省く。大抵の本で最初に出ているのは最急降下法というものである。とりあえず、関数 f が N 次元ユークリッド空間全体で定義されているとする。この方法の原理は、「ちょっと動いた時にもっとも関数値が減る方向に動く」というのを繰り返すことである。もっとも減る方向は、

$$\Delta f = \nabla f \cdot \Delta \boldsymbol{x} \tag{1}$$

とテイラー展開の 1 次までとって、 $|\Delta \boldsymbol{x}|$ が一定の時に $|\Delta f|$ を最大にすればよく、もちろん $\Delta x = \alpha \nabla f$、つまり一階導関数自体が方向ということになる。 これで、あとは前節で述べた適当な方法を使ってその方向での 1 次元最適化を適当にやり、あとはまたそこで新しく勾配を求めて同じことを繰り返す。

最急降下法のよいところは、いつかは収束することであり、よくないところは収束が必ずしも速くないことである。これは、2 変数で目的関数が 2 次形式の場合についていろいろ実験したり考えて

みたりすればわかるが、結局直交する 2 方向の繰り返しに陥ってしまうからである。この事情は多変数の場合でも同じで、結局 2 方向になっていくということが証明されている。

というわけで、速く答を出そうとするなら、やはり疑似ニュートン法的なものを考えたくなる。1 変数の時のように簡単に 2 次形式の近似式が構成できるわけではないが、その辺をなんとかうまくやろうというわけでいろんな方法が提案されている。

多分良くつかわれているのは DFP (Davidon-Fletcher-Powell) 法や BFGS(Broyden-Fletcher-Goldfarb-Shanno) 法で、どちらも考案者の名前である。時々 Variable Metric method (可変計量法) と呼ばれることもある。

以下、基本的な考え方を説明する。

目的関数が、以下の 2 次形式

$$f(x) = \frac{1}{2}x^T Q x \tag{2}$$

であるとする。 x は n 次元実ベクトル、 Q は n 次元正方行列で、対称にとって良いので正定値である。

ニュートン法では、f を 2 次形式で近似してそれの極値を求める。形式的には、近似値 x_0 の回りでのテイラー展開

$$f(x - x_0) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T H(x_0)(x - x_0) \tag{3}$$

の極値を求めるわけである。ここで H はヘッシアンで、2 階偏導関数を要素とする正方行列である。つまり

$$h_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{4}$$

で、極値を与える点は、 $\Delta x = x - x_0$ として、

$$H(x_0)\Delta x = -\nabla f(x_0) \tag{5}$$

である。

目的関数が上の 2 次形式なら、これはもちろん $\Delta x = -Q^{-1}Qx_0 = -x_0$ で、正しい答が求まる。

そういうわけで、考え方としては、ヘッシアンなり Q の近似値を作っていこうというのが基本になる。

ここで、ちょっと定義を続ける。まず、共役という概念を定義する。2 つのベクトル u, v が Q について共役であるとは

$$u^T Q v = 0 \tag{6}$$

であることである。

さらに、互いに共役な k 個のベクトル $p_0, ...p_{k-1}$ を考え $(k < n)$、これらに対して、

$$H_k = \sum_{i=0}^{k-1} \frac{p_i p_i^T}{p_i^T Q p_i} \tag{7}$$

を考えると、明らかに

$$H_k Q p_i = p_i \quad (i = 0, 1, ..., k-1) \tag{8}$$

である。

これは、 H_k が $p_0, ..., p_{k-1}$ が張る部分空間では Q^{-1} みたいなものであるということを意味している。したがって、p_i を順番に発生させる方法がなにかあれば、それを使って、

$$H_{k+1} = H_k + \frac{p_k p_k^T}{p_k^T Q p_k} \tag{9}$$

で H_k を計算していけばよい。

ここでの実際的な問題は、互いに共役な p_k を作るよい方法があるかどうかよくわからないことである。一つの考え方は、反復による修正量 $s_k = x_{k+1} - x_k$ を使うことである。 s_k は共役ではないが、それでもなんとか

$$H_{k+1} Q s_k = s_k \tag{10}$$

が成り立つように、式 (9) を適当に変更する。変更のために右辺に C_k という項を追加することにすると、 これの満たすべき式は

$$(H_k + C_k) y_k = 0 \tag{11}$$

ここで y_k は

$$y_k = Q s_k = \nabla f(x_{k+1}) - \nabla f(x_k) \tag{12}$$

である。

このように C_k をとる一つの方法 (DFP 法) は、

$$C_k = -\frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} \tag{13}$$

とするものである。

まあ、自分でプログラムを書くならこれをつかうので OK であろう。ライブラリとかだと BFGS のほうが少しよいようである。

3.3 CG 法

さて、さっき共役な p_k を作る方法はよくわからないと書いたが、これは原理的には知られている。但し、いつでも上手くいくわけではないのでそれを使わない方法も研究されているわけである。

共役な p_k を直接作る方法が共役勾配法、すなわち CG(Conjugate gradient) 法である。これは形式的には簡単である。いま、$g_k = \nabla f(x_k)$ と書くことにして、

$$p_0 = -g_0 \tag{14}$$

$$p_k = -g_k + \beta_k p_{k-1} \quad (k = 1, 2, ...) \tag{15}$$

というベクトル列を考える。ここで β_k は

$$\beta_k = \frac{g_k^T Q p_{k-1}}{p_{k-1}^T Q p_{k-1}} \tag{16}$$

である。

f が 2 次関数の時には、これで求まる p_k は互いに共役であることを証明できる。さらに、ちょっと変形すると、

$$\beta_k = \frac{|g_k|^2}{|g_{k-1}|^2} \tag{17}$$

となって、これは簡単に計算できる。

なお、疑似ニュートン法、CG 法のどちらの場合でも、方向は決まるが 1 次元問題を繰り返し毎に解く必要はある。これは黄金分割なり疑似ニュートン法なりを使うことになる。多次元の反復では勾配を求めないといけないので少なくとも n 個の関数を計算することになるが、1 次元問題では反復毎に 1 度 f を計算するだけなのでここではそれほど効率に気を使う必要はないことに注意。

3.4 CG 法の応用:連立 1 次方程式

CG 法は現在最適化よりも大規模な線型方程式を解くのに広く使われている。今、

$$Ax = b \tag{18}$$

という線型連立方程式を考える。で、A は対称行列であるとする。この時、上の方程式は、以下の 2 次形式

$$f(x) = \frac{1}{2}x^t Ax - bx \tag{19}$$

を最小化するものと考えることができる。ここで CG 法を使うというのが基本的な考え方である。CG 法なので 1 次元方向の最小化がいるが、これは線型問題なので答がわかっている。

何故こんなややこしいことをするのかと思うかもしれない。理由はちゃんとあって、一つはこの方法は反復法であるにもかかわらず原理的には有限回で収束すること、つまり、ガウスザイデルや SOR とは違って、(計算精度が無限に高いなら) 収束が保証されていることである。もう一つはいろいろ工夫することで収束を非常に速くできることが多いということである。

収束を速くするための工夫は色々な「前処理」と言われるもので、それだけを扱った本がいっぱいあるのでここではこれ以上は扱わない。

4 制約つき最適化

大抵の問題では、目的関数の定義域が実数全体ということはない。例えば、銀河の回転曲線から質量分布を求めようというときには、質量はどこでも正でなければならない。

このような制約にはいろんな場合があるが、制約が等式の場合はラグランジュの未定乗数法が使えるのでこれは省略する。不等式の場合は話が難しくなる。

問題によっては厳密にできる場合もあるが、ここではペナルティ法というものを紹介しておく。これは、

$$g_i(x) \geq 0, \quad i = 1, 2, ..., n \tag{20}$$

という制約のもとで $f(x)$ を最小化せよという問題を、 f と g_i を適当に組み合わせて作った関数を制約なしで最小化せよという問題に置き換える。具体的には、$\{g_i(x)\}$ を、

$$\{g_i(x)\} = \begin{cases} 0, & (g_i(x) \geq 0), \\ g_i(x), & \text{otherwize} \end{cases} \tag{21}$$

つまり、制約条件を満たせば 0 、そうでなければ 0 でないような関数として

$$F(x) = f(x) + p\sum\{g_i(x)\}^2 \tag{22}$$

を最小化する。で、答が求まる度にパラメータ p の値を適当に大きくしていって、こちらの条件に見合う解になったら止める。

この方法では、制約条件のところに解があるとそれに境界の外側から近付く。このために外点法と呼ばれる。内点法というのもあって、これは F を

$$F(x) = f(x) + p\sum \frac{1}{g_i(x)} \tag{23}$$

とする。こちらでは、 p を小さくするにしたがって領域の内側から真の解に近付く。

5 次週予告

今週は、基本的に性質がよい関数に対する決定論的な手法を説明した。来週は、もうちょっとなんだかわからないものにも使える (こともある) 確率的手法を紹介する。

6 練習

プログラムが必要なものはプログラムを提出すること。

1. 1 変数関数
$$f(x) = -xe^{-x^2} \tag{24}$$
の区間 $[0, 2]$ での最小値を黄金分割法で求めるプログラムを作り、答を求めよ。

2. 上の関数について、3 点を使う疑似ニュートン法のプログラムを作り、収束が一次より速いことを確認せよ。

3. 2 変数の 2 次形式
$$f(x) = 0.5x_1^2 + 5x_2^2 \tag{25}$$
について、最急降下法がどのように収束するかをいくつかの適当な初期値について図示せよ。プログラムを書いても手で計算してもよい。

4. 式 (16) が互いに共役なベクトル列を与えることを証明せよ。

5. 問題 3 の収束がどうなるかを CG 法の場合についてしらべよ。