

## [オーガナイズド講演] 確率モデルと集団最適化入門

### Introduction to statistical models for populational optimization

赤穂 昭太郎\*

Shotaro Akaho

**Abstract:** This paper provides an introductory review of populational search methods based on statistical modelling. We start with a random search algorithm called Markov Chain Monte Carlo (MCMC). To solve an optimization problem, it is crucial for the performance to use specific knowledge about the problem. We discuss how the optimizer can acquire the knowledge during the optimization, which is closely related to various fields of statistical learning theory.

## 1 まえがき

学習や推定、適応といった情報処理のさまざまな場面において、最適化問題を解く必要に迫られることは多い。しかも、往々にしてそれらの最適化問題は手に負えない問題である。本解説では、そのような難しい問題が与えられたときに、集団を用いたランダム探索によって解を求める方法を一般的な視点から概説し、基本的な知見をまとめる。一言で言えば、対象となる関数全体の構造を、複数の探索点を使って学習しながら、獲得した知識を用いた最適化を行うことによって効率化を行うという手法であり、学習研究との関連も深い。

最後の手段としての最適化法 はじめに注意しておくが、本稿で紹介する手法は非常に一般的で、どんな問題にも適用可能な夢のような手法であるというイメージをいただくかもしれない。しかしながら、一般性の裏返しとして探索の効率は非常に悪い。従ってまず、もっと簡単な問題に帰着できないかを検討した上で、どうしてもだめな場合に「最後の手段」として用いるべきである。それでも単純に適用するとうまく動かないことが多いので、対象に依存した工夫をいろいろ凝らすことは研究対象になりうる。

理論解析の対象として 一方、IBIS の参加者であれば、最適化問題を解くというよりは、そのアルゴリズムの収束性などの理論的な解析に興味があるかもしれない。確

率論や確率過程の解析対象として面白いだけでなく、最後の手段としての手法に性能保証を与えることは重要である。

## 2 確率的最適化の基本的な枠組み

本稿で述べる集団最適化は確率的最適化 (stochastic optimization) と集団探索 (population search) という二つの側面をもつが、まず確率的最適化について基本的事項をまとめる。

### 2.1 関数最小化問題

なんらかの探索空間  $\mathcal{X}$  上に定義された実数値関数  $f(x) \in \mathbb{R}, x \in \mathcal{X}$  の最小値およびそのときの  $x$  の値  $x^*$  を求めることが問題である。

$$x^* = \arg \min_x f(x) \quad (1)$$

$\mathcal{X}$  や  $f(x)$  にはとりあえず何も制限は設けず、できる限り一般的な枠組みで議論を進める。

関数の種類 一般的な枠組みとは書いたが、関数は大まかに言って以下のように分けることができる。

(1)  $f(x)$  の関数形が既知かどうか:  $f(x)$  が  $x$  の陽な関数形で書けていれば、その関数形の情報を使うことができるので最適化が簡単になる可能性がある。一方  $f$  が完全にブラックボックスで  $x$  を与えてみないと  $f(x)$  がわからないような問題はより難しいと思われる。 $f(x)$  を得るために計算コストがどれくらいかかるかも、最適化の戦略に影響を与えるだろう。

(2) 観測ノイズがあるかどうか: 上のブラックボックスの場合にさらに  $f(x)$  そのものではなく、それにノイ

\*産業技術総合研究所 脳神経情報研究部門, 305-8568 茨城県つくば市梅園 1-1-1 中央第2, e-mail akaho@m.iece.org, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Central 2, Umezono, Tsukuba, Ibaraki 305-8568

ズが付加された  $y = f(x) + \epsilon$  のみが観測されるような問題では、同じ  $x$  でも毎回返ってくる関数値が異なるため、最適化問題としてはより難しいと考えられる。

もちろん、関数形が既知であっても、十分複雑な関数ならば未知でノイズありの場合と似たようなものになってくる。以下の議論ではこれらの分類にあまり依存しない形で解説を行うが、どのような問題であるかによって手法に向き不向きがあるので、注意する必要がある。

## 2.2 ランダム探索

問題にほとんど何も制限がないので、とりあえずランダムに  $x \in \mathcal{X}$  を発生させ、 $f(x)$  が小さな値を取るような  $x$  を選ぶことにしよう。できれば、小さな値を取る  $x$  を優先的にサンプルする方が効率がよい。そこで、

$$p_T(x) = \frac{1}{Z_T} \exp(-f(x)/T) \quad (2)$$

という確率分布に従って  $x$  を生成することにする<sup>1</sup>。exp の上に乗せたのは確率値を正の値にするため、 $Z_T$  は正規化定数

$$Z_T = \sum_{x \in \mathcal{X}} \exp(-f(x)/T) \quad (3)$$

で、式の形を見ればわかるように、一般に  $Z_T$  を計算するのは大変である。 $T > 0$  は温度と呼ばれ、 $T$  が 0 に近づくにつれ、最小値を取る  $x$  でピークを持つ分布になる。

$p_T(x)$  が  $x$  の事後分布のとき、 $f(x)$  の最小値を求める問題は最大事後分布推定 (MAP = Maximum a posterior estimation) を行っていることになる。

問題は  $p_T(x)$  に従う乱数を発生させるのに、 $f(x)$  に関して多くの知識を必要とすることである。これは、 $f(x)$  に関しては何も制限を設けないと仮定したのに反する。この問題を解決するための一つの方法は、以下で述べるように、 $p_T(x)$  に収束するマルコフ連鎖を構成することである。

## 2.3 MCMC 法

ここでは最も一般的なメトロポリス・ヘイスティングス法 (Metropolis-Hastings) の形でマルコフ連鎖モンテカルロ (MCMC = Markov Chain Monte Carlo) 法を定義する。具体的には、極限分布が  $p_T(x)$  に収束するマルコフ連鎖  $P(x_{t+1} | x_t)$  を以下のように構成する。

[メトロポリス・ヘイスティングス法]

1. まず、 $x_t$  から  $x_{t+1}$  の「候補」を条件付確率  $q(x_{t+1} | x_t)$  に従ってランダムに生成する。
2. 次に、 $x_{t+1}$  を以下の確率に従って「採択」し、採択されない場合は「棄却」する (つまり  $x_{t+1} = x_t$  とする)

$$\alpha(x_t, x_{t+1}) = \frac{p_T(x_{t+1})q(x_t | x_{t+1})}{p_T(x_t)q(x_{t+1} | x_t)} \quad (4)$$

ただし  $\alpha(x_{t+1}, x_t) \geq 1$  の場合は必ず採択する。

候補を選ぶ分布  $q$  (提案分布: proposal distribution) はよほど特殊なものを除きかなり広い範囲で選ぶことが可能だが<sup>2</sup>、これをどのように設計するかによって最適化問題を解く効率が左右される。

なお、メトロポリス・ヘイスティングス法の特殊な場合として、ギブスサンプラー (Gibbs sampler) やメトロポリス法 (Metropolis method)、独立サンプラー (independence sampler) などが位置づけられる。これらは  $q$  として特別なクラスのものを考えていることに相当する。

## 2.4 MCMC 法的基本性質

ここでは、MCMC 法に関する重要な性質をまとめておく。より詳細な解説は [8, 12, 15] などの参考書を参照されたい。

**収束性** まず、上で述べた MCMC 法は、適当な条件下で任意の初期値  $x_0$  からスタートしても必ず目標の極限分布  $p_T(x)$  に収束することが示されている。

任意の初期値からの収束性に関する条件は理論的には少し難しいが、少なくとも  $p_T(x)$  が MCMC の定めるマルコフ連鎖  $P(x_{t+1} | x_t)$  に関して定常になることは、詳細つりあい条件 (detailed balance condition) と呼ばれる

$$p_T(x)P(y | x) = p_T(y)P(x | y) \quad (5)$$

が任意の  $x, y$  について成り立つことから容易に示せる。

**局所計算** MCMC 法を実行するためには (4) 式を計算する必要があるが、この中の  $p_T(x)$  は値そのものではなく、 $p_T(x_{t+1})$  と  $p_T(x_t)$  の比だけがわかればよい。これは  $p_T(x)$  を (2) 式のように定めていることに対して非常に都合がよい。(2) 式には正規化定数  $Z_T$  が含まれているが、比を取るとそれが消えてしまうために計算する必要がないということである。従って、MCMC は  $x_t$  と  $x_{t+1}$  だけから定まる関数の計算によって実行可能である。

<sup>1</sup>このためには  $\mathcal{X}$  が可測空間で  $f$  は  $Z$  が定義可能なものである必要があるなどの条件が入るが詳細は省略する。

<sup>2</sup>基本的には確率過程がエルゴード的になるものである必要がある

### 3 確率的最適化の効率化

#### 3.1 提案分布の設計

提案分布の設計が肝だと言ったが、それではどのような設計方針に従えばよいのだろうか。

**Exploration-Exploitation** MCMC 法を用いた最適化法ではお互いに両立しがたい二つの要請がある。一つは、最適化問題を効率的に解くために、できるだけ広い空間をむらなく探索したいということである。そのためには提案分布はできるだけ  $\mathcal{X}$  上を広く飛び回るものである必要がある (Exploration)。

一方で、MCMC 法では採択と棄却というステップが入るため、候補が採択されなければ意味がない。採択されやすい候補というのは単純に言えば現時点の  $x_t$  よりも確率値ができるだけ高いもの (つまり  $f(x_t)$  の値が小さいもの) である。これは  $f(x)$  に何も仮定を入れなければどうしようもないが、たとえば  $f(x)$  に弱い連続性のようなものを仮定すると、今まで  $f(x)$  の値が小さかったところの周りはやはり値が小さいと思われるので、その周りに集中して候補を出すようにすれば、高い確率で採択されるかもしれない (Exploitation)。しかし、それでは最初のできるだけ飛び回って欲しいという要請には矛盾してしまう。

**事前知識の利用から適応的獲得へ** exploration と exploitation を両立するために、もし関数に関する事前知識があればそれを用いることができる。例えば、上に書いたように  $\mathcal{X}$  に位相が入っていて、 $x_t$  の近傍にある点の  $f(x)$  は  $f(x_t)$  に近い値をとるという知識があったとすると、 $q(x_{t+1} | x_t)$  は近傍の点にジャンプするような確率分布を構成することが有効となる。exploration をするためには、近傍系をできるだけ大きく取ればよいが、逆に大きく取りすぎると事前知識の範囲からはみ出してしまふ。

それ以外にも、例えば関数が滑らかで凹な関数であるときに、微分 (離散の場合は差分) に関する情報が使えれば、その勾配を下る最急降下方向に重点的に候補を出せばよい。もっとも、厳密にその条件が成り立つならば MCMC の必要もないので、あくまで大まかな事前情報というぐらいのイメージである。

こうして考えていくと、事前情報が得られない場合にも、関数に関する知識をサンプルからの「学習」によって獲得することによって、それを提案分布の設計に生かすことができないかというのが、EDA などに至る流れであると考えられるが、その話は一旦後回しにする。

#### 3.2 シミュレーテッドアニーリング

さて、もう一つの重要なパラメータとして (2) 式に現れた温度  $T$  がある。すでに述べたように  $T$  が 0 に近づくほど、 $f(x)$  の最小値が強調される凹凸の激しい分布になる。しかしながら、MCMC のように  $\mathcal{X}$  を渡り歩くような方法にとっては、一般に最適値まで到達するのが難しくなる。逆に、 $T$  を大きくすれば、そのような問題は少なくなるが、 $f(x)$  とは無関係にランダムウォークするのと変わらなくなってしまう。

そこで、少しずつ温度を下げていきながら上のような問題を解決しようというのがシミュレーテッドアニーリング (SA=Simulated Annealing) である。このとき問題となるのは、どれくらい速く温度を下げて大丈夫かということである。温度を変えることによって分布が変わってしまうので、平衡状態への収束は阻害されてしまう。従って、平衡状態に十分収束させながら少しずつ温度を下げる必要がある。これについては多くの理論的な成果がある [5, 6]。

**収束定理** (Hajek[6]) 有限状態のシミュレーテッドアニーリングによる系列  $\{x_t\}$  が最適解に収束するための必要十分条件は、 $t$  回目の繰返しで用いる温度を  $T_t$  として、

$$\sum_{t=1}^{\infty} \exp(-D/T_t) = \infty \quad (6)$$

となることである。ただし  $D$  は問題によって定まる定数。

具体的には  $T_t = D/\log t$  はこれを満たすが<sup>3</sup>、実質的には遅すぎるので、実際の運用では  $T_t = \alpha^t T_0$  ( $0 < \alpha < 1$ ) のように指数的に減衰させることが多い。また、ここでは有限状態と書いたが無限状態への一般化なども研究されている。

### 4 集団モンテカルロ法

さて、今までは 1 つの  $x$  が  $\mathcal{X}$  上を渡り歩いて最適解を探すというものであったが、複数の  $x$  で探せばもっと効率がよくなるのではないかというのが本稿の主テーマである集団最適化である。まずここでは MCMC を集団で行う集団モンテカルロ法について説明しよう<sup>4</sup>。

最も単純なアイデアは、 $x$  を  $K$  個用意して、それらに対して独立に MCMC を走らせるというものである。これは、 $\mathcal{X}$  に値をとる  $K$  個の確率変数  $x^{(1)}, \dots, x^{(K)}$

<sup>3</sup>ただし一般に  $D$  は未知である。

<sup>4</sup>本稿では拡張アンサンブル法と呼ばれる枠組みで幅広く研究されているもの一端を紹介する。

を考え,

$$p(x^{(1)}, \dots, x^{(K)}) = \prod_{k=1}^K p_T(x^{(k)}) \quad (7)$$

を極限分布とする MCMC として,  $k = 1, \dots, K$  の順に  $x^{(k)}$  を一つずつ更新していく 1 変数 MCMC をやっていることに相当する. その際提案分布は  $q(x_{t+1}^{(k)} | x_t^{(k)})$  をそのまま用いる. この一般化を行う際に以下の 2 つの方向性が考えられる.

(1) 極限分布 (7) の一般化: 同じ形の分布である必要はない. 例えば, (2) 式で, それぞれの  $k$  ごとに温度が違ふ分布  $p_{T_k}(x^{(k)})$  を使って,

$$p(x^{(1)}, \dots, x^{(K)}) = \prod_{k=1}^K p_{T_k}(x^{(k)}) \quad (8)$$

を極限分布とするようにしてもよい.

(2) 1 変数 MCMC から多変数 MCMC への一般化: 提案分布を 1 変数ごと独立ではないものに拡張することが考えられる. といってもそれだけでは少し広すぎるが, 例えば独立な 1 変数 MCMC ステップに加えて  $x^{(j)}$  と  $x^{(k)}$  の値を入れ替えるという操作によって,  $x^{(j)}$  と  $x^{(k)}$  の間の相互作用のような効果を入れることができる.

パラレルテンパリング法 実際には, (8) 式のように温度を変えた分布に対して, 交換操作を導入したのがパラレルテンパリング (parallel tempering) 法と呼ばれる手法である. ただし, 定常分布を不変に保つために, 交換は単純に行えばいいというわけではなく, 適切な確率で行わなければならない.  $x^{(j)}$  と  $x^{(k)}$  を交換する候補として選んだとしよう.  $j$  と  $k$  は任意の確率分布によって選んでもよいが, 実際に交換するかどうかは以下の確率で決める.

$$\beta(x^{(j)}, x^{(k)}) = \frac{p_{T_k}(x^{(j)})p_{T_j}(x^{(k)})}{p_{T_j}(x^{(j)})p_{T_k}(x^{(k)})} \quad (9)$$

ただし,  $\beta(x^{(j)}, x^{(k)}) \geq 1$  の場合は必ず交換する. この交換が全体の極限分布を不変に保つことは, 全変数の MCMC と見てやれば容易に確認できる.

(9) 式を見るとわかるように, 交換は相手の温度に移ったときにやはり高い確率をとっていないと採択される確率が低い. 通常は近い温度をもつ変数同士 ( $x^{(k)}$  と  $x^{(k+1)}$ ) などで行われる.

パラレルテンパリング法は, 温度の高い分布や低い分布を行ったり来たりしながら探索を行う. そのため, シミュレーテッドアニーリングと似た効果を持ち, 温度の下げ方に神経質になる必要もない. そのかわり,  $T_1, \dots, T_K$

をどのように設定するかはそれなりに重要となる. あまり密に配置すると, 計算時間のオーバーヘッドなどが増えて無駄になる上, 温度の高いところと低いところを行き来するのに多くのステップを必要とするために, 最適化の効率も逆に悪くなる可能性もある. 一方, あまりスパースに配置すると前に述べたように交換の採択確率が下がってしまう. そこで, 経験的に (あるいは漸近理論などを用いて), ある程度一定の採択率をもつような温度の配置がいろいろ提案されている [8].

## 5 最小値探索と期待値計算

もともとモンテカルロ法は関数の最小値を探すことよりは, 確率変数の期待値を計算することに主眼があり, 研究もそちらに重点が置かれてきた. 一方, 本稿では最適化に関する部分についてのみ説明してきたので, MCMC 法の一側面だけに偏っている感が否めない. 最小値を探す難しさと期待値を計算する難しさは微妙に異なるが, 比較的ゆるい条件で, それらは以下のように関係している.

定理  $f(x)$  を  $\mathcal{X} \in \mathbb{R}$  上で有界な閉集合の上の関数とする.  $f(x)$  の最小値を取る  $x = x^*$  がただ一つ存在するとき,  $f(x)$  が  $x = x^*$  で連続なら

$$\lim_{T \rightarrow +0} \frac{\int_{\mathcal{X}} x \exp(-f(x)/T) dx}{\int_{\mathcal{X}} \exp(-f(x)/T) dx} = x^* \quad (10)$$

が成り立つ.

従って, 関数の連続性が仮定できれば, 期待値計算のために研究されたさまざまなテクニックが最小値を探すのにも役に立つ可能性はある. なお, モンテカルロ法を用いた具体的な計算法は [14] なども参考になる.

そもそも最小値探索ではピンポイントで最適点だけに関心があるが, 統計的な手法という観点からは, もっとソフト化して, 「ある  $\eta, \delta$  に対して確率  $1 - \eta$  以上で

$$f(x) < f_0 + \delta \quad (11)$$

となる  $x$  が見つかることを確率的に保証する」といった, PAC 学習的な考え方が向いているように思える (ただし  $f_0$  は真の最小値).

### 5.1 重点サンプリング法

MCMC では, ある極限分布  $p(x)$  に関する期待値を計算するための重要な方法に重点サンプリング (importance sampling) 法がある.

今,  $p(x)$  の代わりに,  $r(x)$  に従う  $N$  個のサンプル  $x_1, \dots, x_N$  が得られたとしよう.  $x_i$  は必ずしも独立で

ある必要はない。一般に期待値を計算するためのサンプルは独立である必要はないという事実は重要である<sup>5</sup>。MCMC のように時間の前後でサンプル間に相関があってもちゃんと期待値を計算できるのもそこに理由がある。さて、この  $N$  個のサンプルに重み  $w_i = p(x_i)/r(x_i)$  を定義して、重み付き平均を計算すると、

$$\langle x \rangle_w = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} \quad (12)$$

は  $N$  が大きくなるに従って、確率変数  $x$  の  $p(x)$  に関する期待値に確率収束する。

重点サンプリング法を使えば、必ずしも  $p(x)$  に従うサンプルが得られない場合でも  $w_i$  さえ計算できれば  $p(x)$  に関する期待値を計算できる。さらに、 $w_i$  は相対的な大きささえわかればよいので、この場合も正規化定数  $Z$  の計算は必要ない。

一般に MCMC は収束するまでに時間がかかるので実時間計算が必要な時系列フィルタリングなどの用途にはそのままでは適さないが、重点サンプリングを使うことにより対処できる。実際、逐次モンテカルロ法 (sequential Monte Carlo) あるいはパーティクルフィルタ (particle filter) [3, 7] と呼ばれる手法では、重点サンプリングと次節のリサンプリングを組み合わせることによって高速にフィルタリングを行う方法を与える。

## 5.2 リサンプリング

重点サンプリングの際に用いた  $r(x)$  に従うサンプル  $x_1, \dots, x_N$  を  $w_i$  に比例する確率

$$\tilde{w}_i = \frac{w_i}{\sum_{i=1}^N w_i} \quad (13)$$

で復元抽出するリサンプリング (resampling) を実行して新たな  $N$  個のサンプル集合  $x'_1, \dots, x'_N$  を作る。  $x_1, \dots, x_N$  が独立ならば  $N$  が大きくなるに従って、各  $x'_i$  は  $p(x)$  に法則収束する。

リサンプリングでは、 $w_i$  の大きいサンプルのコピーを作り、 $w_i$  の小さなサンプルは消滅させるという「分裂・消滅過程」である。これはいわゆる次元の呪い (curse of dimensionality) を避けたり、 $p(x)$  に従うサンプルが得られたりするなど、いくつかのメリットはあるものの、有限サンプルではサンプルのバリエーションが減ってしまうためあまり乱用はしない方がよい。

## 5.3 遺伝的アルゴリズム

遺伝的アルゴリズム (GA = Genetic algorithm) は DNA の進化などにヒントを得て考え出された集団最適

<sup>5</sup>ただし、正の相関が強いと期待値の推定量の分散が大きくなってしまう。

化法の一つであるが、ここでは今まで述べてきた方法との関連を明らかにしておく。

遺伝的アルゴリズムは基本的には、以下の  $N$  個の集団に対して、以下の 3 つの操作を施すことによって定義される確率過程である。

1. 選択: 確率  $p_T(x_i)$  に従ってリサンプリングを行う。
2. 突然変異: ある確率分布  $q(x_{t+1} | x_t)$  に従って各  $x_t$  から  $x_{t+1}$  を作る (通常は微小ノイズを加える)。
3. 交差: ある確率分布  $r(x_{t+1}^{(j)}, x_{t+1}^{(k)} | x_t^{(j)}, x_t^{(k)})$  に従って二つの個体  $x_t^{(j)}, x_t^{(k)}$  から新たな個体  $x_{t+1}^{(j)}, x_{t+1}^{(k)}$  を作る (例えば  $x$  が文字列のときそれらを途中で切って「交差」させる)。

さて、まず「選択」については、前節で見たリサンプリング法と同様の手続きである。ただし、 $r(x) = 1$  にしたもになっているので、現在の集団が  $p_{T_t}(x)$  に従うものであったとすると、 $p_T(x)$  でリサンプリングすることによって得られる集団の分布は

$$p_{T_{t+1}}(x) = p_{T_t}(x)p_T(x) = \exp(-f(x)/T_{t+1})/Z_{t+1}, \quad (14)$$

$$T_{t+1} = 1/(1/T_t + 1/T) = T_t/(1 + T_t/T) < T_t \quad (15)$$

という分布になる。つまり、集団の温度の値はどんどん低くなり、リサンプリングと同時にシミュレーテッドアニーリングを行っていることになる。リサンプリングとシミュレーテッドアニーリングの効果により、集団は単一化する傾向にあるが、これは遺伝的浮動 (genetic drift) と呼ばれる。探索の多様性を保つためにはこれを避けたほうがよいこともある [1]。

次に「突然変異」については MCMC 法の中で提案分布に従って候補を生成するステップとみなせる。候補の採択・不採択はないが、分布への収束は「選択」におけるリサンプリングで行うということである。

最後の「交差」は遺伝的アルゴリズムに特有のステップである。4 節において、1 変数 MCMC から多変数 MCMC への拡張について言及したがそれを具体化したものと言える。二つの個体を使っているという点ではパラレルテンパリング法と同じだが、パラレルテンパリング法では交換のみを行っていた<sup>6</sup>。ただし、実際に採択率の高い「交差」の確率過程がどんなものになるかは与えられた問題の性質に大きく依存するはずであり、その辺りの議論が必要となるであろう。

<sup>6</sup>この場合すべての個体は同じ温度の上で動いているので交換をしても意味はない

## 6 確率構造探索と最適化

3.1 節などいろいろなところで述べたように、最適化アルゴリズムの設計において、解くべき問題の構造を反映させることはパフォーマンス向上に有効である。事前知識が十分に得られない場合は、最適化の段階で適応的に問題の構造を獲得しながら、それに基づいて最適化を行うことになり、学習・適応といった研究分野との関連性が深くなっていく。近年、このような観点から、遺伝的アルゴリズムを中心とした分野において EDA (Estimation of Distribution Algorithm) と呼ばれる手法が活発に研究されるようになってきた。EDA については本オーガナイズドセッションの残りの講演で詳しい解説があるので、本稿では EDA については最小限にとどめ、関連すると思われる研究について「構造獲得と最適化」という観点から考察する。

はじめに注意しておく、あくまで効率的に最適化をすることが目的なので、問題の構造を学習するための計算コストはそれほど大きくできないことがある。あまり凝った学習法をやって時間をかけているうちに、単純な手法に負けるといった事態になりかねない。

### 6.1 問題構造とは何か

今まで「問題構造」という漠然とした用語を使ってきたが、ここでは統計モデル、つまり統計的な学習の対象となるような構造のことを指すことにする。見かけ複雑な構造をした関数も、内在的に十分簡単な (= 統計的学習が可能な) 構造を持っているという仮定に立った立場である。

ここで、何を学習するかについていくつかの自由度がある。一つは MCMC などが対象とする確率分布  $p_T(x)$  そのもののモデル化が考えられる。

$$p_T(x) \simeq p(x; \theta) \quad (16)$$

ここで  $\theta$  はパラメータである。 $p_T(x)$  そのものに従うサンプルを得るのは難しくても  $p(x; \theta)$  が  $p_T(x)$  を十分よく近似し、かつサンプリングをするのが簡単な分布であれば、その分布を学習することによりサンプリングが楽になるだろう。後で述べる平均場近似を用いた手法は基本的にはこの立場である。

ただし、 $p_T(x)$  そのものは複雑過ぎて単純な分布では近似できないことも多い。そのような場合に、最適化のためには  $p_T(x)$  全体を近似する必要はなく、 $f(x)$  が小さな値をとるような  $x$  がもれなくサンプリングされればよいという考え方も出来る。 $p_T(x)$  全体を近似するのは難しくても、 $f(x)$  が小さいところに限定すれば比較的少ないパラメータの分布で記述可能かもしれない。ま

た、 $p_T(x)$  が小さいところが MCMC のランダムウォークの障壁になっている場合には、これを忠実に実現するよりは多少の下駄を履かせて障壁を除くようにしたほうがよいようにも思える。

EDA は基本的に後者のように  $f(x)$  の小さいところに限定してサンプルを得ようという考え方であり、そのナイーブな実現法として以下のようなものが考えられる。

[ナイーブな EDA]

1. サンプル集合  $x_t^{(1)}, \dots, x_t^{(N)}$  から  $f(x)$  が小さいものを選ぶ
2. 選ばれたサンプルを使って分布  $p(x; \theta)$  を学習する。
3.  $p(x; \theta)$  に基づいて新たなサンプル集合を作る。

EDA は遺伝的浮動の傾向など GA との共通部分も多い。実際には MCMC との統合や、具体的な  $p(x; \theta)$  のクラスなどが問題となるが、本稿ではそれらの問題点には踏み込まず、以下、学習に関する側面について関連する研究を紹介する。

### 6.2 能動学習

通常統計的な学習では、学習サンプルは受動的に与えられるのみで、そこから統計的な構造を抽出するという枠組みで行われる。つまり、未知の確率分布  $p(x)$  に従う  $x_1, \dots, x_N$  が与えられたときに、 $p(x)$  を推定するという問題である。一方、能動学習 (Active learning) [4] というのは、サンプルのうち学習者がコントロール可能な部分がある学習であり、より効率的に学習を行える可能性がある。例えば、 $x \in \mathcal{X} = [0, 1]$  上の  $y = ax + b + \epsilon$  という 1 次関数の学習 ( $\epsilon$  はノイズ) で、学習者が  $x$  を指定できるとしよう。このとき ランダムに  $x$  を選ぶよりは、 $x = 0, 1$  という端の点を選んだほうが  $a, b$  の推定精度は明らかに高くなる。この問題は最適実験計画 (optimal experimental design) と呼ばれる分野で古くから研究されてきた。

本稿の文脈で見れば、探索点  $x \in \mathcal{X}$  は自由に選ぶことが出来るので、能動学習によって、少ないサンプル点から問題の確率的な構造を学習することが可能になると期待される。

上の例では「1 次関数である」というような、モデル族に関する知識の存在が前提として必要となる。ただし一般にはそのような知識はないことが多いので、モデル族そのものの探索も必要となる。これは学習理論の研究者なら誰でも知っているように、モデル選択という難しい問題である。能動学習の分野でもモデル選択と能

動学習を同時に行うという研究はなされているが、実際に適用して能動学習の効果が得られるようにすることは容易ではない。

### 6.3 bandit problem と強化学習

能動学習は統計モデルを効率的に同定する目的のみでサンプリングが行われる。本稿での主目的はあくまで最適化なので、モデルの同定と最適化を両立する必要がある。そのような方向性において関連が深いものの一つに bandit problem がある。これはコインの出る確率が異なるスロットマシンがいくつかあるときに最大の収益を上げたいという問題である。

この問題では、各スロットマシンの統計的性質の同定と、収益最大化という二つの異なる目的を達成する必要がある。一般的には、最初のうちは統計的性質を同定するために使って、その後決めたスロットを引き続けるという最適戦略がよく知られている。

さらにこれを一般化した問題が強化学習 (RL = Reinforcement learning) [16] である。強化学習で考えるのは、bandit problem でスロットを引く試行が各回独立ではなく、マルコフ的に状態遷移するような状況である。このような枠組みはマルコフ決定過程 (MDP = Markov decision process) と呼ばれ、確率パラメータが既知ならば動的計画法 (DP = Dynamic programming) を利用した再帰計算による効率的な解法が知られている。しかし通常は確率パラメータが未知なので、統計モデルの同定をしながら最適な行動を選択する Q-learning や TD-learning といったいくつかの手法が知られている。ただし、これらの枠組みは最適化問題としてかなり特殊なものなので、一般の最適化問題にどのように利用できるかは未知の部分が多い。

一つ参考になるのは、これらの枠組みにおいても、3.1 節で説明した exploration-exploitation が重要な要素となっていることである。確率モデルを同定するためには exploration を行って幅広いサンプルを集める必要があるし、高い収益を上げるには既存の知識に基づいた exploitation によって行動する必要がある。

もう一つの重要な点は、強化学習などでは必ずしも統計モデルのパラメータそのものを同定するとは限らない点である。肝心なのは大きな収益を上げることだから、行動決定に必要な情報のみの学習をするのである。これによって、通常統計モデルの同定に必要なサンプル数よりもかなり少ない数のサンプルからの学習が可能になっている。ただし、これも一般の最適化問題の場合にどこまで通用するかは未知である。

### 6.4 平均場近似

さて、複雑な確率分布を近似するためのものとして、平均場近似 (MFA = Mean field approximation)[13, 10] に基づく手法がある。目的の確率分布がマルコフランダム場 (MRF = Markov random field) あるいはベイジアンネットワーク (BN = Bayesian network) といったグラフィカルモデルによって記述されているとしよう。 $p_T(x)$  自体がその形をしている場合もあるし、EDA のように問題構造を獲得するためのモデルとして利用することもできる。

グラフィカルモデルではよく知られているように単連結なグラフ (singly-connected graph) では確率値計算が容易で、最大値や期待値の計算も動的計画法による再帰計算が可能である (Viterbi アルゴリズムなどがそれに当たる)。しかしながら、ループのあるグラフ (loopy-graph) では、なんらかの近似が必要となる。

確率変数が  $x = (x_1, \dots, x_K)$  という  $K$  個の成分を持つとしよう。最も単純なナイーブ平均場近似は、確率分布  $p(x)$  を

$$p(x) \simeq \prod_{i=1}^K p_i(x_i; \theta_i) \quad (17)$$

という形の分布で近似し、最も  $p(x)$  に近くなるようにパラメータ  $\theta_i$  を決める (変分ベイズ法と呼ばれるものも基本的には同じ)。各変数の周辺分布としての  $p_i(x_i; \theta_i)$  はかなりよい場合が多いが、同時分布として見ると大きくずれている場合が多いと言われている。

ナイーブ平均場近似よりも精度を上げた近似としてベーテ (Bethe) 近似というものがあるが、これは単連結グラフの場合に用いられる確率伝播法 (Belief-propagation) をループのあるグラフに局所的に適用したものに一致している。このアルゴリズムを  $p_T(x)$  に対して適用すると、 $p_T(x)$  の最大値 (つまり  $f(x)$  の最小値) を探索するアルゴリズムに用いることができる。 $p_T(x)$  がある条件を満たせばループのあるグラフであっても最大値を見つげられることがわかっている [18]。

## 7 その他の話題

ここでは確率最適化や集団探索において今まで述べられなかったいくつかの話題について触れておく。

### 7.1 変数変換と潜在変数

最適化においては定義域である  $x \in \mathcal{X}$  の表現が重要であり、単純な変数変換によって問題が簡単になることがある。変数変換と一口に言ってもさまざまなものがあるが、多くの問題を難しくしているのは多数の確率変数の

間の相互作用である。もし変数変換によって各変数が独立にできれば、問題は格段に易しくなる。実数空間の場合、最も基本的な手法は主成分分析 (PCA = Principal component analysis) による無相関化である。また、さらに独立性を追求する手法として独立成分分析 (ICA = Independent component analysis) を用いることも有望かもしれない。離散値の空間では、単純に PCA をかけたりすることはできないが、近似的に離散値を実数空間に埋め込んで PCA や ICA を行ったり、指数分布族 PCA[2]などを適用することは可能である。

また、拡張アンサンブル法の一つであるマルチカノニカル (Multicanonical) 法は、関数値の空間でモンテカルロ法を行うという意味で一種の変数変換ともとらえられる。マルチカノニカル法は集団探索ではないが、問題の構造の学習も行っているという点本稿の手法と関連が深い。

さて、変数変換と類似の手法として、隠れ変数あるいは潜在変数という補助変数  $z$  を導入し、

$$p(x) = \sum_z p(x, z) \quad (18)$$

となっているとみなせるとしよう。  $p(x, z)$  の最大値が簡単に求められるとき、それを利用して  $p(x)$  の最大値を効率的に求める手法が EM アルゴリズム [10] である。

## 7.2 複雑な空間での最適化

確率最適化が必要なほど難しい問題では、 $\mathcal{X}$  が時系列、文字列、木、グラフなど複雑な構造を持ったものであることが多い。このような空間では例えば列の長さやノードの個数といったものが変化するので、確率変数の次元が一定でない。そのような場合もそれら全体の空間に確率構造を入れてやることによって、基本的には同じ枠組みで議論できる。例えば  $\mathcal{Y}$  をアルファベットとする時系列全体は  $\mathcal{X} = \bigcup_{i=1}^{\infty} \mathcal{Y}$  という確率過程全体に確率測度を入れてやればよい。

このような  $\mathcal{X}$  でのランダム探索では次元の異なるところへとジャンプする必要があり、確率変数の生成・消滅という 5.2 節で述べたのと似た過程が現れ、興味深い。このようなジャンプを提案分布として使った MCMC はジャンプ拡散 (jump diffusion) MCMC と呼ばれ研究されている。

## 8 おわりに

確率最適化を中心とした集団探索に関して、学習理論に関連が深い話題を中心にいくつかのアプローチを概説した。詳細については参考文献を参考にしたり、キー

ワードで web 検索などをするとたいいてい必要な情報は得られるだろう。なお、凸性やなめらかさなどきれいな性質をもつ問題に対する「きれいな」最適化問題については、[9, 11] などがよい入門書である。また、複雑な問題を (ランダムネスを入れないで) 近似的にそれらの問題に置き換える方法に関しては [17] が詳しい。

## 参考文献

- [1] S. Akaho. Statistical learning in optimization: Gaussian modeling for population search. In *Proc. of Int. Conf. Neural Information Processing (ICONIP'98)*, 1998.
- [2] S. Akaho. The e-PCA and m-PCA: dimension reduction by information geometry. In *Proc. of Int. Joint Conf. on Neural Networks (IJCNN2004)*, pp. 129–134, 2004.
- [3] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice*. Springer-Verlag, 2001.
- [4] 福水健次. 能動学習の理論, 渡辺ほか: 学習システムの理論と実現. 森北出版, 2005.
- [5] S. Geman and D. Geman. Stochastic Relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on PAMI*, Vol. 6, pp. 721–741, 1984.
- [6] B. Häjek. Cooling schedules for optimal annealing. *Math. Operation. Research*, Vol. 13, pp. 311–329, 1988.
- [7] 樋口知之. 粒子フィルタ. 電子情報通信学会誌, Vol. 88, No. 12, pp. 989–994, 2005.
- [8] 伊庭幸人ほか. 計算統計 II. 統計科学のフロンティア 12. 岩波書店, 2005.
- [9] 茨木俊秀, 福島雅夫. 最適化の手法. 情報数学講座 14. 共立出版, 1993.
- [10] 樺島祥介, 上田修功. 平均場近似・EM 法・変分ベイズ法, 計算統計 I. 統計科学のフロンティア 11. 岩波書店, 2003.
- [11] 金谷健一. これなら分かる最適化数学. 共立出版, 2005.
- [12] J.S. Liu. *Monte Carlo strategies in scientific computing*. Springer-Verlag, 2001.
- [13] M. Opper and D. Saad, editors. *Advanced Mean Field Methods: Theory and Practice*. MIT Press, 2001.
- [14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing, Second edition*. Cambridge University Press, 1992.
- [15] C.P. Robert. *Monte Carlo statistical methods*. Springer-Verlag, 2004.
- [16] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. MIT Press, 1998.
- [17] V.V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001. 邦訳: 浅野孝夫 (訳) 近似アルゴリズム, シュプリンガー・フェアラーク東京, 2002.
- [18] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. on Information Theory*, 2003.