ガウス過程とベイズ最適化

1　ガウス過程

ガウス過程は教師あり学習に分類される手法であり、多くの場合回帰モデルとして利用される。回帰とは一般的に、入力 $\boldsymbol{x} \in V$ に対する出力 $y \in \mathbb{R}$ を関数 $f : V \to \mathbb{R}$ でモデル化する手法である (ここで V は入力の定義域)。ひとたび関数 f が得られれば、教師データ $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)|i = 1, ..., n\}$ にない入力 $\boldsymbol{x}^*$ に対しても、出力の予測値 $f(\boldsymbol{x}^*)$ が得られるようになる。

ガウス過程による回帰はベイズ統計とノンパラメトリックモデルの性質を兼ね備えている点で特徴的である。パラメトリックなベイズ線形回帰モデルの場合、関数 f は $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ のように、$\boldsymbol{w}$ というパラメータを有する。そして $\boldsymbol{w}$ の事後分布 (例えば $p(\boldsymbol{w}|\mathcal{D}_n)$) を計算することで、関数 f のバラつきを求める。ここで言う「関数 f のバラつき」とは、入力 $\boldsymbol{x}$ が与えられたときの、$f(\boldsymbol{x})$ の確率分布 $p(f(\boldsymbol{x})|\boldsymbol{x}) = p(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}|\boldsymbol{x})$ というより、関数の集合 $\{f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}|\boldsymbol{w} \in \mathbb{R}^D\}$ に対応する確率分布である。従って $\boldsymbol{w}$ の事後分布と f の事後分布は実質同じものと考えてよい。

パラメトリックなベイズ機械学習では、パラメータの事後分布という観点で議論するが、ノンパラメトリックモデルの場合は、後者の「関数の分布」という観点の方が分かりやすい。ところで解析学によると、関数 f はベクトルとして表現できた。つまり、全ての $\boldsymbol{x}_i \in V$ に対して、対応する関数値を $f_i = f(\boldsymbol{x}_i)$ とし、それを順不同に並べたベクトル $(f_1, f_2, ...)^{\mathrm{T}}$ も関数の一表現として扱う。このとき、関数の分布とは多次元ベクトルの確率分布として見なせる。

このように、ガウス過程では主に 3 つのベクトルを扱う。一つは入力 $\boldsymbol{x} \in V$ であり、他の回帰モデルと同様である。そして出力 $y \in \mathbb{R}$ であり、実際に対象の系から得られた値である。本資料ではこれを単に出力や観測値と呼ぶことにする。出力をベクトルとして解釈する場合は、関数のベクトルと同様に、$\boldsymbol{y} = (y_1, y_2, ...)^{\mathrm{T}}$ とする。ここで、y_i は入力 $\boldsymbol{x}_i$ に対応する観測値である。3 つ目のベクトルは前述の通り回帰モデルに関するベクトル $\boldsymbol{f}$ である。$\boldsymbol{f}$ の成分について、$\boldsymbol{x}_i$ に対する関数値 $f(\boldsymbol{x}_i)$ を単に f_i と書くことにする。

1.1　ガウス過程の基礎

定義 1.1 ガウス過程

関数 $f : V \to \mathbb{R}$ が、$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, K)$ に従うとき、f のことをガウス過程と言う。ここで $\boldsymbol{\mu} = (\mu(\boldsymbol{x}_1), \mu(\boldsymbol{x}_2), ...)$ であり、関数 $\mu : V \to \mathbb{R}$ のことを平均関数と言う。また、$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ であり、関数 $k : V \times V \to \mathbb{R}$ によって定められる。f がこのガウス過程に従うことを $f \sim \mathcal{GP}(\mu, k)$ と書き表す。

上記は確率論における定義であるが、機械学習ではガウス過程に従う回帰モデルのことを単にガウス過程と言う。

ガウス過程の例としてパラメトリックなベイズ線形回帰を考えよう。いま、線形回帰モデル $f(\boldsymbol{x}) = \sum_{i=1}^{H} w_i\phi_i(\boldsymbol{x})$ に対し、$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \lambda^2 I)$ なる事前分布を考える。ここで $\phi : V \to \mathbb{R}$ は基底関数であり、今回は H 個の基底関数を考えている。従って $\boldsymbol{w} \in \mathbb{R}^H$ である。集合 V を $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ...\}$ のように書いたとき、この線形回帰モデルは

$$\begin{bmatrix} f_1 \\ f_2 \\ \cdots \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \cdots & \phi_H(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_2) & \phi_2(\boldsymbol{x}_2) & \cdots & \phi_H(\boldsymbol{x}_2) \\ \cdots & & & \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_H \end{bmatrix}$$

のように書き表される。よって $\boldsymbol{f}$ は $\boldsymbol{w}$ を線形写像したもので、かつ $\boldsymbol{w}$ はガウス分布に従うため、$\boldsymbol{f}$ の事前分布も $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \lambda^2\Phi\Phi^{\mathrm{T}})$ に従うことが分かる。ここで Φ は計画行列である。従ってこのベイズ線形回帰モデルは、$\lambda^2\Phi\Phi^{\mathrm{T}}$ と整合する関数 k によって定められたガウス過程 $\mathcal{GP}(\boldsymbol{0}, k)$ と言える。

上式のようにベイズ線形回帰では基底関数とパラメータを明示的に考えるが、これは時に計算コスト上問題となる。ベイズ線形回帰のパラメータ数は H であったことを思い出そう。H とは基底関数の数であり、多いほど表現力の高い回帰モデルとなる。基底関数としてガウス基底関数がよく用いられる (図 1(a))。厳密でない表現になるが、図 1(b) から想像できるように、ガウス基底関数が”密に配置”されるほど表現力の高い回帰モデルとなりそうである。すると、入力 $\boldsymbol{x}$ の次元 D が大きくなるほど、各軸方向に対して密に基底関数を配置する必要が出てくるので (図 1(c))、入力次元と共に必要な H の数も指数関数的に増加することが想像できる。これは正に一種の次元の呪いであり、計算コスト上問題になる理由である。

一方で、本章で紹介するガウス過程は基底関数の代わりに関数 k を考える。機械学習の分野では k のことをカーネル関数

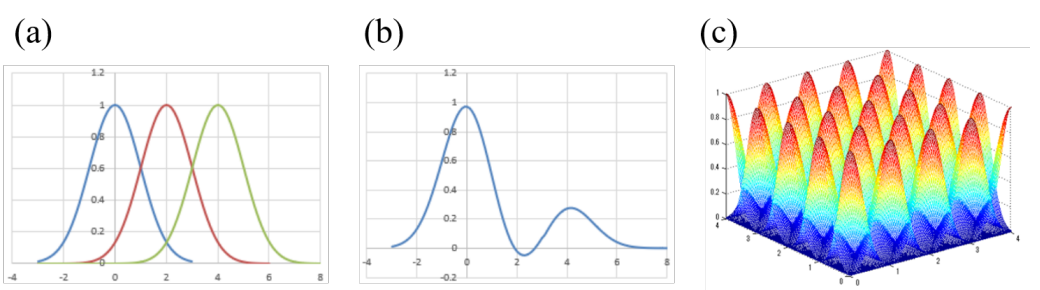

図 1.1　(a) ガウス基底関数 (b) ガウス基底関数を用いた場合の線形回帰モデル (c)2 次元入力に対するガウス基底関数

と言う。これまでの議論より、関数の事前分布として知りたいのは Φ ではなく $\lambda^2\Phi\Phi^{\mathrm{T}}$ であることに気付く。入力 $\boldsymbol{x}$ が連続値である場合、V の元の数は無限であるため、$\Phi \in \mathbb{R}^{\infty\times H}$ となる。そのため $\lambda^2\Phi\Phi^{\mathrm{T}} \in \mathbb{R}^{\infty\times\infty}$ となる。行と列の数が無限になることは置いておくとして (ガウス過程ゆえに解析的な処理が容易なので、計算に困らない)、この行列のサイズが H に依存しなくなったことに注目したい。従って、基底関数を定義するのではなく、カーネル関数から直接的に分散共分散行列を定義した場合、H の増加による次元の呪いを回避できる。

以上がガウス過程の基本的な前提である。このようにガウス過程では基底関数を明示的に考えないので、分散共分散行列も $\lambda^2\Phi\Phi^{\mathrm{T}}$ ではなく単に K と書く。このような行列をカーネル行列と言い、以下のように定義される。

$$K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{1.1}$$

1.1.1 ガウス過程のベイズ統計的側面

先程のベイズ線形回帰モデルは平均関数がゼロのガウス過程という特殊な例であった。平均関数値が常にゼロなのは、暗黙的に存在するパラメータ $\boldsymbol{w}$ の期待値をゼロとしたためである。

本資料の残りでも平均関数はゼロの定常関数とする。実は教師データ $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)|i = 1, ..., n\}$ に対して観測値 y_i の平均がゼロとなるように前処理を行えば、平均関数がゼロであっても上手くいくというのが広く知られている。そもそもゼロ以外の平均関数を設定するということは、$\boldsymbol{w}$ の期待値を非ゼロに設定することに相当する。これはモデルに対して前提となる知識を有している場合なら上手く働くかもしれないが、あまりそういった機会には出会わない (前提知識を有しているならばパラメトリックな回帰モデルを用いるべきであるため)。従って、ガウス過程の事前分布は以下の通りである。

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{0}, K). \tag{1.2}$$

次に、回帰モデルに対する観測値の尤度を考える。一般的に回帰モデルと観測値の間に

$$Y = f(\boldsymbol{x}) + \epsilon \tag{1.3}$$

なる関係式を考えることが多い。ここで ϵ は観測誤差に相当する確率変数であり、本資料では $\epsilon \sim \mathcal{N}(0, \sigma^2)$ とする。この観測誤差があるゆえに観測値も確率変数として取り扱う。本資料では観測値を Y で表し、その実現値を y とする。式 (1.3) より、尤度は

$$p(\boldsymbol{Y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\sigma^2 I) \tag{1.4}$$

となる。ここで $\boldsymbol{Y} = (Y_1, Y_2, ...)$ である。これは条件付確率であるため、$\boldsymbol{f}$ は実現値であることに注意してほしい (本来なら別の表記にすべきであるが、これが慣習である)。

ところで私たちが知りたいのは関数の分布であるが、それは式 (1.2) の事前分布ではなく事後分布であり、つまり $p(\boldsymbol{f}|\boldsymbol{Y}_n = \boldsymbol{y}_n)$ である。ここで $\boldsymbol{y}_n = (y_1, ..., y_n) \in \mathbb{R}^n$、$\boldsymbol{Y}_n = (Y_1, ..., Y_n) \in \mathbb{R}^n$ である。ただし、特に知りたいのは教師データにある入力 $X_n = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ 以外の入力 $X^* = \{\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, ...\}(X_n^c = X^*)$ であろう。そこで関数 $\boldsymbol{f}$ を $\boldsymbol{f} = (\boldsymbol{f}_n, \boldsymbol{f}^*)^{\mathrm{T}}$ のように分割して書くことにする。なお $\boldsymbol{f}_n = (f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_n))^{\mathrm{T}} \in \mathbb{R}^n$、$\boldsymbol{f}^* = (f(\boldsymbol{x}_1^*), f(\boldsymbol{x}_2^*), ...)^{\mathrm{T}}$ である。このとき、$(\boldsymbol{f}_n, \boldsymbol{f}^*)^{\mathrm{T}}$ の事前分布は

$$p\left(\begin{bmatrix}\boldsymbol{f}_n \\ \boldsymbol{f}^*\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{0}_n \\ \boldsymbol{0}^*\end{bmatrix}, \begin{bmatrix}K_{ff} & K_{f*} \\ K_{*f} & K_{**}\end{bmatrix}\right)$$

のように書き表すことができる。なお、

$$K_{ff}(i,j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad K_{*f}(i,j) = k(\boldsymbol{x}_i^*, \boldsymbol{x}_j), \quad K_{f*}(i,j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j^*), \quad K_{**}(i,j) = k(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*)$$

である。未知の入力に対する事後分布を知りたい場合は、$p(\boldsymbol{f})$ を $\boldsymbol{f}^*$ で周辺化すればよい。つまり、

$$p(\boldsymbol{f}^*|\boldsymbol{y}) = \int p(\boldsymbol{f}|\boldsymbol{y}) d\boldsymbol{f}_n = \frac{1}{p(\boldsymbol{y})} \int p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}) d\boldsymbol{f}$$

の計算を行う (最右辺はベイズの定理より導出)。この積分は以下のように求まる。

$$p(\boldsymbol{f}^*|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}^*, \Sigma^*), \quad \boldsymbol{\mu}^* = K_{*f}(K_{ff} + \sigma^2 I)^{-1}\boldsymbol{y}_n, \quad \Sigma = K_{**} - K_{*f}(K_{ff} + \sigma^2 I)^{-1}K_{f*} \tag{1.5}$$

X^* のうち有限個の入力 $X_m = \{\boldsymbol{x}_1^*, ..., \boldsymbol{x}_m^*\}$ に対する事後分布のみ知りたい場合は、式 (1.5) を更に周辺化すればよい。そこで以下の定理を紹介する。本定理より、カーネル行列の定義を X_n と X_m に関するものに置き換えれば、式 (1.5) と同様であることに気付く。

定理 1.1

多次元ベクトル $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)^{\mathrm{T}}$ が以下の分布

$$\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix} \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

に従うとする。このとき、周辺確率 $p(\boldsymbol{x}_1)$ は $\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_{11})$ である。

1.1.2 カーネル関数

カーネル関数はカーネル行列を決定する重要な関数である。カーネル行列はガウス分布の分散共分散行列となる。分散共分散行列は対称性および半正定値性を有するため、カーネル関数もそれを満たすように設計されていなければならない。つまり、カーネル関数として使えるものは

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i), \quad \sum_{ij} c_i c_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \tag{1.6}$$

を満たす関数に限られる。ここで $c_i(c_j)$ は任意の実数である。以下はカーネル関数に関する自明な定理である。

定理 1.2

関数 k' 及び k'' が式 (1.6) を満たすとき、線形結合 $k = ak' + bk''$ 及び積 $k = k'k''$ も式 (1.6) を満たす。

カーネル行列が共分散行列であるということは、$k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ が f_i と f_j の共分散となることを意味している。従って対象の問題に対して f_i と f_j に相関があるならば、$k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ も大きな値となるように設計されなければならない。多くの場合において回帰モデルは滑らかな関数とするので、$\boldsymbol{x}_i$ と $\boldsymbol{x}_j$ の類似度に合わせて大きな値となるカーネル関数を設定する。

以下はよく用いられるカーネル関数である。

- **線形カーネル**

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

- **指数カーネル**

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta}\right)$$

  ここで θ は正の実数であり、調整可能なパラメータである。θ が小さい程カーネル関数値は小さくなりやすくなる。そのため θ を小さくすることで類似度の判定を厳しくすることができると表現できる。このようなパラメータをスケールパラメータと言う。

- **RBF カーネル**

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^2}{2l^2}\right)$$

  ここで l はスケールパラメータである。

- **周期カーネル**

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\theta_1 \cos\left(\frac{|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta_2}\right)\right)$$

  ここで θ_1, θ_2 はスケールパラメータである。周期カーネルは $\boldsymbol{x}$ の類似度に周期性を持たせている点で特徴的である。

- **Matern カーネル**

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta}\right)^{\nu} K\left(\frac{\sqrt{2\nu}|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta}\right)$$

  ここで θ はスケールパラメータ、ν はカーネルの微分特性を調整するパラメータである。また、K は第二種の変形ベッセル関数である。$\nu \to \infty$ のとき、Matern カーネルは RBF カーネルと一致する。
  RBF カーネルは解析のしやすさなどの理由ゆえに機械学習の分野でよく採用されていた。一方で、RBF カーネルの無限回微分可能という性質は回帰モデルとして極端な前提であることも指摘されている。そこで、可能な微分回数が調整できる Matern カーネルが提案された次第である。Matern カーネルは ν より大きい自然数の内最小の数だけ微分可能である。ν の値としてよく用いられるのが 3/2 と 5/2 で、それぞれ Matern3 および Matern5 と呼ばれる。

$$(\nu = 3/2): \quad k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \frac{\sqrt{3}|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta}\right) \exp\left(-\frac{\sqrt{3}|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta}\right)$$
$$(\nu = 5/2): \quad k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \frac{\sqrt{5}|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta} + \frac{5|\boldsymbol{x}_i - \boldsymbol{x}_j|^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\theta}\right)$$

1.1.3 解析結果例 (sample.py)

本資料作成を通して、ガウス過程のためのモジュール「pyGP」を作成した。pyGP は本資料と同じ場所に保存している。以下は sine 関数に対する回帰モデル例である。なお、観測誤差としてサンプルデータには標準偏差 0.1 のガウス分布に従う誤差を加えた。

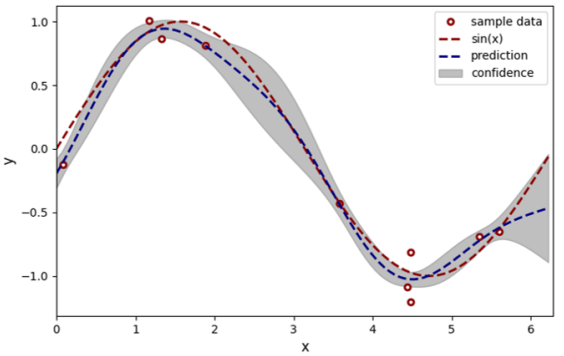

図 1.2　ガウス過程解析結果例

1.2　補助変数法

本節ではガウス過程の計算コスト対策として提案された補助変数法を紹介する。前述の通り、ガウス過程では行列 $K^{nn}+\sigma^2 I$ の逆行列を求めなければならない。これは N 行 N 列の行列であるため、ガウスの消去法だと $\mathcal{O}(n^3)$ の演算コストを要する。従ってビッグデータの場合問題になってくる。また、密行列ゆえにメモリコストも無視できない。

補助変数法は教師データ $\mathcal{D}_n=\{(\boldsymbol{x}_i,y_i)|i=1,...,n\}$ の代わりに疑似的なデータ $\{(\boldsymbol{z}_i,u_i)|i=1,...,l\}$ を考える。ここで $l<n$ である。補助変数法では $\{\boldsymbol{z}_i|i=1,...,l\}$ のことを補助変数と言う。

疑似データのうち u_i は実現値として考えない。それよりも補助変数 $\boldsymbol{z}_i$ をよしなに決めて、それに対応する確率変数 U_i を考える。U_i は $\boldsymbol{z}_i$ における回帰モデルの予測なので、

$$\boldsymbol{U}_l\sim\mathcal{N}(\boldsymbol{0},K^{ll}) \tag{1.7}$$

を満たす。ここで $\boldsymbol{U}_l=(U_1,...,U_l)^{\mathrm{T}}$、$K^{ll}_{ij}=k(\boldsymbol{z}_i,\boldsymbol{z}_j)$ である。

補助変数は入力データ $X=\{\boldsymbol{x}_i|i=1,...,n\}$ から決められる。最も単純な方法として、X からランダムに l 点選択し、それを補助変数として使うことが考えられる (このやり方を SoD 法と言う)。もう少しこだわるならば、クラスタリングの利用が考えられる。例えば K-means 法を用いて、X を l 個のカテゴリにクラスタリングし、各分類の代表点 (例えば重心) を補助変数として利用する。

補助変数法はガウス過程の近似手法であり、近似のさせ方が異なる幾つかの手法の総称である (SoD 法はそのうちの一つ)。本資料では SoR と DTC 法、並びに FITC 法を紹介する。補助変数法全体において共通している近似は、$\boldsymbol{U}_l$ に対する条件付き独立性である。いま、関数 $\boldsymbol{f}$ を $(\boldsymbol{f}_n,\boldsymbol{f}^*,\boldsymbol{U}_l)^{\mathrm{T}}$ のように分けて表記する。このとき、

$$p(\boldsymbol{f}_n,\boldsymbol{f}^*)=\int p(\boldsymbol{f}_n,\boldsymbol{f}^*|\boldsymbol{U}_l)p(\boldsymbol{U}_l)d\boldsymbol{U}_l$$

が成立する。また、$p(\boldsymbol{f}|\boldsymbol{U}_l=\boldsymbol{u}_l)$ と $p(\boldsymbol{f}^*|\boldsymbol{U}_l=\boldsymbol{u}_l)$ に関しては式 (1.5) より

$$\begin{aligned}p(\boldsymbol{f}|\boldsymbol{U}_l=\boldsymbol{u}_l)&=\mathcal{N}(K_{fu}K_{uu}^{-1}\boldsymbol{u}_l,K_{ff}-Q_{ff})\\ p(\boldsymbol{f}^*|\boldsymbol{U}_l=\boldsymbol{u}_l)&=\mathcal{N}(K_{*u}K_{uu}^{-1}\boldsymbol{u}_l,K_{**}-Q_{**})\end{aligned}$$

と厳密に求まる。ここで

$$Q_{ab}\equiv K_{au}K_{uu}^{-1}K_{ub} \tag{1.8}$$

である。一方で、補助変数法は $p(\boldsymbol{f}_n,\boldsymbol{f}^*|\boldsymbol{U}_l)$ の計算を効率化するために、近似を施す。特に重要な仮定は $\boldsymbol{U}_l$ に対する独立性であり、つまり、$p(\boldsymbol{f}_n,\boldsymbol{f}^*|\boldsymbol{U}_l)=q_n(\boldsymbol{f}_n|\boldsymbol{U}_l)q^*(\boldsymbol{f}^*|\boldsymbol{U}_l)$ のように近似する。q^* 及び q_n の定め方は手法によって異なる。

1.2.1　SoR 法

SoR 法は $\boldsymbol{U}_l$ と回帰モデルの間に

$$\boldsymbol{f}_n=K_{fu}K_{uu}^{-1}\boldsymbol{u},\quad \boldsymbol{f}^*=K_{*u}K_{uu}^{-1}\boldsymbol{u}$$

なる線形性を仮定する。このとき、$\boldsymbol{f}_n$ と $\boldsymbol{f}^*$ の同時確率分布は

$$p(\boldsymbol{f}_n,\boldsymbol{f}^*)\simeq\int q_n(\boldsymbol{f}_n|\boldsymbol{U}_l)q^*(\boldsymbol{f}^*|\boldsymbol{U}_l)p(\boldsymbol{U}_l)d\boldsymbol{U}_l=\mathcal{N}\left(\boldsymbol{0},\begin{bmatrix}Q_{ff}&Q_{f*}\\Q_{*f}&Q_{**}\end{bmatrix}\right)$$

のように求まる。以上より、$\boldsymbol{f}^*$ の事後分布は式 (1.5) より

$$q(\boldsymbol{f}^*|\boldsymbol{y})=\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*),\quad \mu^*=\sigma^{-2}K_{*u}\Sigma K_{uf}\boldsymbol{y}_n,\quad \Sigma^*=K_{*u}\Sigma K_{u*} \tag{1.9}$$

となる (あくまで近似手法なので $p(\boldsymbol{f}^*|\boldsymbol{y})$ ではなく $q(\boldsymbol{f}^*|\boldsymbol{y})$ と書いた)。ここで

$$\Sigma=(\sigma^{-2}K_{uf}K_{fu}+K_{uu})^{-1} \tag{1.10}$$

である。式 (1.9) を用いて予測をすればよい。以上より、SoR 法は l 行 l 列の逆行列を計算すれば済むこと、並びに n 行 n 列の行列を定義しなくてもよいことが分かる。

図 3(a) に解析結果を示す。平均値は概ねよい結果を示しているものの、分散を過少に評価していることが分かる。

1.2.2 DTC 法

DTC 法は SoR 法と同様に線形性を仮定するものの、$p(\boldsymbol{f}^*|\boldsymbol{U}_l)$ は厳密な確率分布を用いる。従って、$\boldsymbol{f}_n$ と $\boldsymbol{f}^*$ の同時確率分布は

$$p(\boldsymbol{f}_n, \boldsymbol{f}^*) \simeq= \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} Q_{ff} & Q_{f*} \\ Q_{*f} & K_{**} \end{bmatrix}\right)$$

となる。このときの事後分布は

$$q(\boldsymbol{f}^*|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \quad \mu^* = \sigma^{-2} K_{*u} \Sigma K_{uf} \boldsymbol{y}_n, \quad \Sigma^* = K_{**} - Q_{**} + K_{*u} \Sigma K_{u*} \tag{1.11}$$

となる。平均値は SoR 法のときと変わらない一方で、分散共分散行列は $K_{**} - Q_{**}$ だけ足されている。証明はしないが $K_{**} - Q_{**}$ は正定値行列であるため、DTC 法による結果の分散は SoR 法のときより大きい。解析結果を図 3(b) に示す。

1.2.3 FITC 法

FITC 法は DTC 法の同時確率分布に更なる改良を加える。具体的には Q_{ff} のうち対角成分のみ K_{ff} のものに置き換える。つまり、

$$p(\boldsymbol{f}_n, \boldsymbol{f}^*) \simeq= \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} Q_{ff} - \mathrm{diag}(Q_{ff} - K_{ff}) & Q_{f*} \\ Q_{*f} & K_{**} \end{bmatrix}\right)$$

とする。このときの事後分布は

$$q(\boldsymbol{f}^*|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \quad \mu^* = \sigma^{-2} K_{*u} \Sigma K_{uf} \Lambda^{-1} \boldsymbol{y}_n, \quad \Sigma^* = K_{**} - Q_{**} + K_{*u} \Sigma K_{u*} \tag{1.12}$$

となる。ここで

$$\Lambda = \mathrm{diag}(K_{ff} - Q_{ff} + \sigma^2 I) \tag{1.13}$$

である。DTC よりも計算コストは高いが、それでも n 行 n 列の逆行列の計算を回避している (Λ は対角行列であるため容易に逆行列計算可能)。解析結果を図 3(c) に示す。

2 ベイズ最適化

ベイズ最適化とは、数学的に定義された関数の最適解を見つける手法であり、いわゆる最適化手法に分類されるものである。最適化手法には様々な種類があるが、ベイズ最適化は対象の関数にこれといった性質を要求しない (例えば勾配を用いた最適化の場合、関数の微分可能性を要求する)。このような性質をブラックボックス関数に対応可能であると言う。

ブラックボックス関数に対応可能な点に着目すれば、ベイズ最適化は進化計算と同類である。しかしながら、ベイズ最適化は対象の関数をベイズ回帰モデル (特にガウス過程回帰) で近似する点で、一般的な進化計算手法と異なる。本資料では、ガウス過程 $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, K)$ でモデル化した場合に限定して議論する。

ガウス過程でモデル化するメリットとして、任意の入力 $\boldsymbol{x} \in V$ における観測値の期待値だけでなく、その自信度合いも定量的に評価できる点にある。後述するように、ベイズ統計は最適化の間逐次的にデータ $(\boldsymbol{x}, y)$ を能動的サンプリングして、最終的に最適解が得られるよう設計されている。このプロセスにおいて活用と探索の機構は、他の手法と同様に極めて重要となる。平均関数に注目することは活用に相当し、逆に分散に注目することは探索に相当する。ベイズ最適化は、平均関数 $\mu(\boldsymbol{x})$ と分散共分散行列を用いて定義された関数 $\alpha : V \to \mathbb{R}$ を指針にして次のサンプル点を決め、最適解へ近づいていく。このような関数を獲得関数と言う。

ここまでの話を踏まえて、以下にベイズ最適化のアルゴリズムを示す。

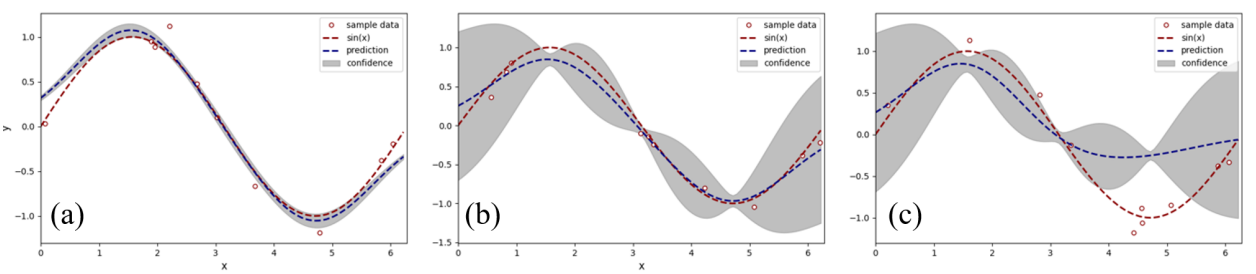

図 1.3 ガウス過程解析結果例。(a)SoR 法 (b)DTC 法 (c)FITC 法。

ベイズ最適化のアルゴリズム

1. 初期条件として、n 個のデータをサンプリングする。つまり、$\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)|i = 1, ..., n\}$ を得る。なお n の値は任意である。
2. D_n を用いて式 (1.5) 等のガウス過程回帰モデルを構築する。
3. 2. のガウス過程回帰モデルに従い、獲得関数が最大となる入力 $\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x}\in V}\alpha(\boldsymbol{x})$ を求める。
4. $\boldsymbol{x}_{n+1}$ についてサンプリングし、観測値 y_{n+1} を得る。
5. $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\boldsymbol{x}_{n+1}, y_{n+1})\}$ と更新する。
6. 終了条件を満たしていない場合、$n = n + 1$ とし、2. に戻る。
7. 最終的に得られたデータ $\mathcal{D}_n$ のうちの $X_n = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ に対し、期待関数が最大となる $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in X_n}\mu_n(\boldsymbol{x})$ を最適解として出力する。

獲得関数については様々なモデルが提案されており、その内のいくつかを次節で紹介する。アルゴリズム中 3. の $\boldsymbol{x}_{n+1}$ を求めるにも、最適値探査が必要となる。これは、獲得関数が微分可能である場合、勾配を用いた最適値探査などで求める。

本資料では終了条件について触れない。様々な手法が提案されているが、本資料中の例題はいずれも反復回数の上限を終了条件としている。

2.1 獲得関数

2.1.1 改善確率量獲得関数

データ $\mathcal{D}_n$ が得られた時点における暫定的な最適値 y_n^* に対し、任意の入力 $\boldsymbol{x} \in V$ に対する出力 y がそれよりも最適となる確率を考える。例えば最小値探査の問題である場合、この確率は $\Pr(y < y_n^*)$ となる。このあと直ぐに導出するように、この確率は $\boldsymbol{x}$ の関数となる。この関数が最大となる $\boldsymbol{x}$ は、次のサンプリング点として筋がよい。このような考えに基づく獲得関数を改善確率量獲得関数と言い、以下のように定義する。

$$\alpha(\boldsymbol{x}) = \Pr(y < y_n^*). \tag{2.1}$$

暫定的な最適値 y_n^* は、定義より $y_n^* = \min_i y_i$ と書ける。$\mathcal{D}_n$ が所与のときのガウス過程回帰モデルの事後分布は式 (1.5) で表されるため、任意の $\boldsymbol{x} \in V$ における $\alpha(\boldsymbol{x})$ は

$$\alpha(\boldsymbol{x}) = \int_{-\infty}^{y_n^*} \frac{1}{\sqrt{2\pi\sigma_n^2(\boldsymbol{x})}} \exp\left(-\frac{(y - \mu_n(\boldsymbol{x}))^2}{2\sigma_n^2(\boldsymbol{x})}\right) dy$$

より求まる。ここで μ_n は n 個のデータが与えられた段階の平均関数であり、関数 $\sigma_n^2(\boldsymbol{x})$ は式 (1.5) 中 Σ の $\boldsymbol{x}$ に対応する対角成分である。つまり、

$$\sigma_n^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_{*f}(\boldsymbol{x})^{\mathrm{T}}(K_{ff} + \sigma^2 I)^{-1}\boldsymbol{k}_{f*}(\boldsymbol{x})$$

である。ここで $K_{ff} \in \mathbb{R}^{n\times n}$ は $\mathcal{D}_n$ に関連するカーネル行列であり、また

$$\begin{aligned} \boldsymbol{k}_{*f}(\boldsymbol{x}) &= (k(\boldsymbol{x}, \boldsymbol{x}_1), ..., k(\boldsymbol{x}, \boldsymbol{x}_n))^{\mathrm{T}} \\ \boldsymbol{k}_{f*}(\boldsymbol{x}) &= (k(\boldsymbol{x}_1, \boldsymbol{x}), ..., k(\boldsymbol{x}_n, \boldsymbol{x}))^{\mathrm{T}} \end{aligned} \tag{2.2}$$

である。標準正規分布の累積分布関数 $\Phi : \mathbb{R} \to \mathbb{R}$ を用いることで、

$$\alpha(\boldsymbol{x}) = \Phi\left(\frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) \tag{2.3}$$

を得る。

前章で紹介したベイズ最適化のアルゴリズムより、次のサンプル点 $\boldsymbol{x}_{n+1}$ は式 (1.2) を最大とする $\boldsymbol{x}$ を選べばよい。改善確率量獲得関数は $\boldsymbol{x}$ に関して微分可能であるため、$\boldsymbol{x}_{n+1}$ の探査に勾配法を用いることができる。導出過程は省略するが、勾配 $\boldsymbol{\nabla}\alpha \in \mathbb{R}^D$ (D は V の次元数) の第 i 成分 $\partial_i\alpha \in \mathbb{R}$ は

$$\partial_i\alpha(\boldsymbol{x}) = \phi\left(\frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right)\left(\frac{-\partial_i\mu_n(\boldsymbol{x})\sigma_n(\boldsymbol{x}) + \mu_n(\boldsymbol{x})\partial_i\sigma_n(\boldsymbol{x})}{\sigma_n^2(\boldsymbol{x})}\right) \tag{2.4}$$

より求まる。ここで関数 $\phi : \mathbb{R} \to \mathbb{R}$ は標準正規分布関数である。式中に平均関数 $\mu_n(\boldsymbol{x})$ と $\sigma_n(\boldsymbol{x})$ の勾配も計算する必要があるが、ガウス過程回帰が $\mathcal{N}(\boldsymbol{0}, K)$ で考えられている場合、それぞれ

$$\begin{aligned} \partial_i\mu_n(\boldsymbol{x}) &= (\partial_i\boldsymbol{k}_{*f}(\boldsymbol{x}))^{\mathrm{T}}\left(K_n + \sigma^2 I\right)^{-1}\boldsymbol{y}_n \\ \partial_i\sigma_n(\boldsymbol{x}) &= \frac{1}{2\sigma_n(\boldsymbol{x})}\left(\partial_i k(\boldsymbol{x}, \boldsymbol{x}) - \partial_i\boldsymbol{k}_{*f}(\boldsymbol{x})^{\mathrm{T}}\left(K_n + \sigma^2 I\right)^{-1}\boldsymbol{k}_{f*}(\boldsymbol{x}) - \boldsymbol{k}_{*f}(\boldsymbol{x})^{\mathrm{T}}\left(K_n + \sigma^2 I\right)^{-1}\partial_i\boldsymbol{k}_{f*}(\boldsymbol{x})\right) \end{aligned} \tag{2.5}$$

となる。

以上より、改善確率獲得関数を用いた場合のベイズ最適化が導出できた。改善確率獲得関数は最も基本的な関数の一つであり、比較的高速に計算できることが分かっている (データ数 n に対して $\mathcal{O}(n^2)$。ただしカーネル行列の逆行列計算は除く)。一方で y_n^* 以下となる確率に注目することは、活用に重きを置きすぎているという指摘もあり、実際に最適化が上手くいかない問題も多い。また、観測誤差が大きい場合も不得意としていることが知られている。

2.1.2 期待改善獲得関数

本項で紹介する期待改善獲得関数は、経験的に改善確率獲得関数よりも上手く最適化できることが知られている。期待改善獲得関数は $\alpha(\boldsymbol{x}) = \mathbb{E}(\max(0, y_n^* - y))$ のように定義されている。y_n^* に着目する点は改善確率獲得関数と同様だが、期待改善獲得関数は確率ではなく差分で定義する。

期待改善獲得関数は解析的に求めることができ、

$$\alpha(\boldsymbol{x}) = (y_n^* - \mu_n(\boldsymbol{x}))\Phi\left(\frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) + \sigma_n(\boldsymbol{x})\phi\left(\frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) \tag{2.6}$$

である。また、この勾配の第 i 成分は以下の通りである。

$$\partial_i\alpha(\boldsymbol{x}) = -\partial_i\mu_n(\boldsymbol{x})\Phi\left(\frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) + \partial_i\sigma_n(\boldsymbol{x})\phi\left(\frac{y_n^* - \mu_n(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})}\right) \tag{2.7}$$

2.1.3 信頼下限獲得関数

信頼下限獲得関数は直感的に分かりやすく理論的な考察も容易であるため、多くの書籍で紹介されている。一方で改善確率獲得関数とは逆に探索圧力が強すぎると言われており、実用上の利用頻度は低い。

信頼下限獲得関数は

$$\alpha(\boldsymbol{x}) = -\mu_n(\boldsymbol{x}) + \sqrt{\beta_n}\sigma_n(\boldsymbol{x}) \tag{2.8}$$

のように定義されている。ここで β_n はハイパーパラメータである。信頼下限獲得関数を最大化する方向に新しい点を探索することで、平均関数値が低くてかつ分散が大きい点をサンプリングするようになる。

β_n の決め方は任意であるが、$\beta_n = (\log n)/n$ とすることが多い。ここで n は既にサンプリングされたデータの数である。こうすることで、サンプリングされるにつれ活用の圧力を高めることができる。なお、信頼下限獲得関数は探索圧力が強いと指摘している書籍では、$\beta_n = c\log n$ としていた。ここで c は実数である。この場合、β_n は n と共に増加していくため、$\beta_n = (\log n)/n$ のときと期待する意味合いが真逆である。$c\log n$ の場合に探索圧力が強いのは納得できる。一方で $(\log n)/n$ のときも探索圧力が強いと言えるのかについて私は知らない。

信頼下限獲得関数の微分は以下の通りである。

$$\partial_i\alpha(\boldsymbol{x}) = -\partial_i\mu_n(\boldsymbol{x}) + \sqrt{\beta_n}\partial_i\sigma_n(\boldsymbol{x}) \tag{2.9}$$

2.2 制約付きベイズ最適化

本節では制約付きのベイズ最適化を考える。制約付き最適化の問題設定は様々だが、本資料では

$$\min f(\boldsymbol{x}), \quad \text{s.t.} \;\; c_i(\boldsymbol{x}) \leq 0 \;\; (i = 1, ..., m) \tag{2.10}$$

に統一する。$c(\boldsymbol{x}) \geq 0$ の制約を扱いたい場合は $c(\boldsymbol{x})$ の符号を逆転すればよい。また、等式制約は $c(\boldsymbol{x}) \leq 0$ かつ $c(\boldsymbol{x}) \geq 0$ と考える。そのため、上式のみを考えても一般性を欠くことはない。なお、等式制約について $c(\boldsymbol{x}) \leq 0$ かつ $c(\boldsymbol{x}) \geq 0$ というのは時に厳しすぎるため、代わりに $c(\boldsymbol{x}) \leq \delta$ かつ $c(\boldsymbol{x}) \geq \delta (\delta \neq 0)$ を考えることもある。

ブラックボックス関数を扱うベイズ最適化では、関数 c もブラックボックス関数であることが多い。この場合、ハードな制約を最適値探査時に課すことは難しい。そのため、制約を満足しない解もサンプリングするような手法 (ソフトな制約) がベイズ最適化では好まれる。

本節で紹介する制約付き最適化の最も特徴的な点は、ブラックボックスな制約関数 c_i に対してもガウス過程を適用することである。つまり、

$$c_i \sim \mathcal{GP}(0, k_i)$$

なる関係を考える。ここで k_i は制約関数 c_i のためのカーネル関数である。c_i に対応する観測値を z^i とし、観測誤差の分散値を σ'^2 とする。このとき、z^i の確率分布は $z^i \sim \mathcal{N}(c_i(\boldsymbol{x}), \sigma'^2 I)$ に従う。

入力点 $\boldsymbol{x}_i$ に対し、観測値として y_i だけでなく $\{z_i^1, ..., z_i^m\}$ も得られる。従って教師データは $\mathcal{D}_n = \{(\boldsymbol{x}_i, ..., y_i, z_i^1, ..., z_i^m) | i = 1, ..., n\}$ のように書き表される。

2.2.1 制約付き期待改善獲得関数

本項では制約付き獲得関数の例として制約付き期待改善獲得関数を紹介する。

まず 2.1.2 項と比べ、y_n^* の選び方が異なる。制約付き期待改善獲得関数においてその選び方は複数あり、Gardner らは

$$y_n^* = \min\{y_i, |z_i^j \leq 0, \quad i = 1, ..., n \quad j = 1, ..., m\} \tag{2.11}$$

なる選び方、つまり制約を満たすデータの内最小となるものを選ぶ方法を提案している。

次に、獲得関数を

$$\alpha(\boldsymbol{x}) = \mathbb{E}\{\Delta(\boldsymbol{x})\max(0, y_n^* - y_i)\} \tag{2.12}$$

のように定義する。ここで $\Delta(\boldsymbol{x})$ は $\boldsymbol{x}$ において全制約を満たす確率である。$c_j(\boldsymbol{x})$ に対応する観測値 z^j が制約を満たす確率は

$$\int_{-\infty}^{0} p(z^j | \mathcal{D}_n) = \Phi\left(-\frac{\mu_n^j(\boldsymbol{x})}{\sqrt{(\sigma_n^j(\boldsymbol{x}))^2 + \sigma'^2}}\right)$$

となる。ここで $\mu_n^j(\boldsymbol{x})$ はガウス過程 $c_j(\boldsymbol{x})$ の平均関数値、$\sigma_n^j(\boldsymbol{x})$ は標準偏差値である。一般的に各制約のガウス過程は独立に従うと仮定されるので、$\Delta(\boldsymbol{x})$ は上式を全制約に対して積を取ったもの、つまり

$$\Delta(\boldsymbol{x}) = \prod_{j=1}^{m} \Phi\left(-\frac{\mu_n^j(\boldsymbol{x})}{\sqrt{(\sigma_n^j(\boldsymbol{x}))^2 + \sigma'^2}}\right) \tag{2.13}$$

となる。

以上より制約付き期待改善獲得関数を定義できた。あとはベイズ最適化のアルゴリズムに従って計算を進めればよい。