

## 機械学習で楽しむ JAX/NumPyro v0.1.2

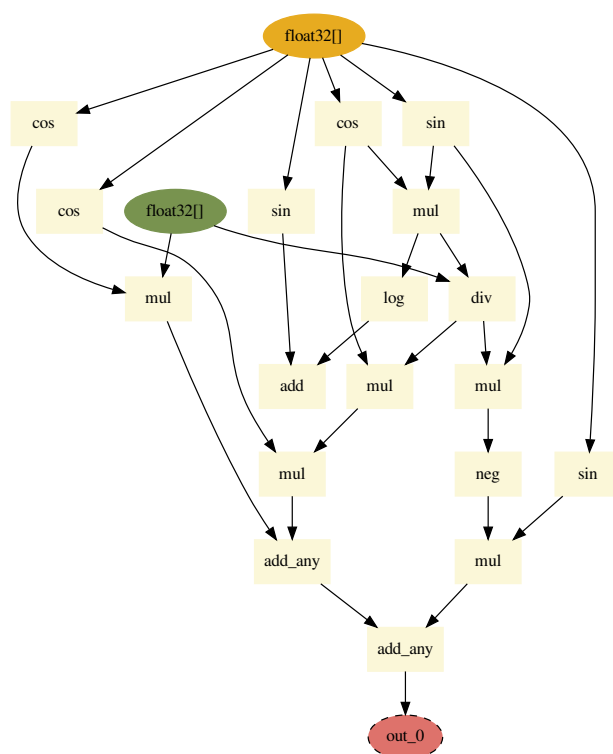


図1  $\partial_x (\log \cos x \sin x + \sin x)$  の計算グラフ

河原創

© 2020–2022 Hajime Kawahara

2022年10月10日

c		
cic	自動微分の簡単な紹介	
ci	確率プログラミング言語	I
	T, †	v
ic	自動微分	v
i	Hbqループを自動微分する	c{
i{	TRy とクラス	cD
iJ	自動微分のカスタマイズ	c_
ii	U†i^~\ e%のインデックス操作	c
{	T, †	{
{ic	U†iC†eCqS C^z Yi	{
{i	T, † bez	v
J	] ~\ e%φ	u
Jic	曲線フィット	u
Ji	カスタマイズした自動微分で ^~\ e%φ を動かす	{u
I	] ~\ e%φ	JCE
Iic	ガウス過程	JCE
Ii	ガウス過程ノイズを含んだモデルフィット	J_
v	] ~\ d%φ	Ic
vic	s<e%w, †w] ~\ d%φ でサンプリング	Ic
,	yRlr	I{
, ic	, qfS	I{
3	, eeC^@	II
3ic	, i ガウス分布	II

## C

T, † は LbblXC が開発中の自動微分と † X, f- <<CCq zC@XS^G q, YL4q g パッケージです<sup>1c</sup>。つまり自動微分+線形代数パッケージで、なおかつ ^-\ e%& 同じ形式で書けるようになっています。本稿はこの T, † と ddX として特に ^-\ e%&。また最適化ツール T, † bez を使って遊びたいというための文書です。普通であれば「T, † w^-\ e%& で学ぶ機械学習」のようなタイトルになると思うのですが、むしろ私は機械学習の問題で T, † w^-\ e%& で遊びたいのです。というのも、毎年、年末年始になると時間ができるので新しい言語やパッケージで遊びたくなるのです。|Cfc 年はコロナ下ということもあり T, † で太陽系外惑星のスペクトル推定コードを書いてみようというのをやってみて査読論文にしました。随時更新する予定です<sup>1c</sup>。

本稿で用いられているコードは LSP-4 においたのでご利用ください。

## cic

勾配ベースの最適化や  $O[\ ; Q ] yr$ 、変分推定にはモデルを最適化あるいは推定したいパラメタで微分できる必要がある。数値的に微分値を求めるにはおおまかに4つの方法がある。一つ目は手で微分  $f \setminus \wedge \sim Y$  @S CqC^zS^S^g をし、結果をコーディングすることである。これはモデルが複雑になってくると破綻しがちであり、またフレキシビリティの観点からもモデルの継続的な改良を妨げる。次に  $\setminus -zPC \setminus -zS-$  等による  $s\% \circ 4bS- @S CqC^zS^S^g$  を用いて手で微分する代わりに微分結果を得てコーディングすることが考えられる。これは  $\setminus -zPC \setminus -zS-$  等を用いたことのある方ならばわかると思うが、モデルが複雑になってくると膨大な項数の結果が排出されるので、やはりフレキシビリティの観点からは難点がある。次に、数値微分を行うことが考えられる。数値微分はモデルが複雑になってくるとエラーがたまりやすい。そこで機械学習分野などで用いられているのが T, † でも用いられている自動微分である fGLi Q:g コーディングできるモデルは通常、様々な微分の既知である関数（ここでは要素関数と呼ぼう）の加減乗除の組み合わせでできている。そこで、自動微分では、連鎖則が成り立つように各要素関数

$$x \rightarrow f(x) \quad \text{fcicg}$$

を関数のアウトプットとその微分値の情報を出力するように拡張する。また微分の加乗除の演算規則がなりたつように演算を定義する。自動微分の実装法としては T, † では T- <bfs^, C-zbq dqb@-z fT, dg を用いたものが使用されている。しかしここではイントロダクションとしてより説明が容易な、双対数を用いた方式を使

<sup>1c</sup> 質問・コメント・お問い合わせは @sfqz2L \ -Sf<b\ までお寄せください。なお筆者はソプラノサクソプレイヤー兼野生の天文学の研究者です。

用して自動微分を解説しよう。

双対数  $z \in k[\epsilon]/\langle \epsilon^2 \rangle$  は  $a, b \in \mathbb{R}$  にたいし、

$$\begin{aligned} z &= a + b\epsilon && \text{fci|g} \\ \epsilon^2 &= 0 && \text{fci\{g} \end{aligned}$$

となる数である。複素数は  $i^2 = -1$  であったのが  $\epsilon^2 = 0$  となったと考えればよい。変数  $x$  の拡張として実部  $a$  に  $x$  を、非実部  $b$  に  $x'$  を割り当てると、 $z = f + f'\epsilon, w = g + g'\epsilon$  に対し、可算・乗数・除算はそれぞれ

$$\begin{aligned} z + w &= (f + g) + (f' + g')\epsilon && \text{fciJg} \\ zw &= fg + (f'g + fg')\epsilon && \text{fci|g} \\ z/w &= \frac{f'g - fg'}{g^2} && \text{fci\vg} \end{aligned}$$

となる。これは実部に関して通常の、また非実部に関して微分の加乗除則と同じであることを示している。次に連鎖則を実現するには関数  $x \mapsto F(x)$  を

$$x + x'\epsilon \mapsto F(x) + F'(x)x'\epsilon \quad \text{fciug}$$

と拡張すればよい。ここで  $F(x)$  を元の関数、双対数をもつ拡張された関数を  $\hat{F}(x + x'\epsilon)$  と分けて表記する。すると式 **fciug** は

$$\hat{F}(x + x'\epsilon) = F(x) + F'(x)x'\epsilon \quad \text{fciDg}$$

と書ける。さて連鎖則  $G(F(x))$  の拡張は

$$\begin{aligned} \hat{G}(\hat{F}(x + x'\epsilon)) &= \hat{G}(F(x) + F'(x)x'\epsilon) && \text{fci_g} \\ &= G(F(x)) + G'(F(x))F'(x)x'\epsilon && \text{fci\cg} \end{aligned}$$

となり、実部は通常の合成  $fG(F(x))g$  が、非実部は連鎖則  $G'(F(x))F'(x)x' = \frac{dG}{dF} \frac{dF}{dx} x'$  が実現されているのが確認できる。

## cicic

というわけで小さな自動微分を **d%Pb^** で実装してみよう。ここで解きたい問題を

$$F(x) = \mathbf{Y}\mathbf{L}(\langle \mathbf{b}\mathbf{s}\mathbf{x}\mathbf{s}\mathbf{S}\mathbf{x} \rangle) + \mathbf{s}\mathbf{S}\mathbf{x} \quad \text{fciccg}$$

の  $x = 1$  で微分  $F'(1)$  とする。さらにその合成関数

$$\begin{aligned} G(x) &= F(F(x)) \\ &= \mathbf{Y}\mathbf{L}(\langle \mathbf{b}\mathbf{s}(\mathbf{Y}\mathbf{L}(\langle \mathbf{b}\mathbf{s}\mathbf{x}\mathbf{s}\mathbf{S}\mathbf{x} \rangle) + \mathbf{s}\mathbf{S}\mathbf{x}) \mathbf{s}\mathbf{S}(\mathbf{Y}\mathbf{L}(\langle \mathbf{b}\mathbf{s}\mathbf{x}\mathbf{s}\mathbf{S}\mathbf{x} \rangle) + \mathbf{s}\mathbf{S}\mathbf{x}) \rangle) + \mathbf{s}\mathbf{S}(\mathbf{Y}\mathbf{L}(\langle \mathbf{b}\mathbf{s}\mathbf{x}\mathbf{s}\mathbf{S}\mathbf{x} \rangle) + \mathbf{s}\mathbf{S}\mathbf{x}) \end{aligned} \quad \text{fci\cg}$$

の  $x = 1$  での微分  $G'(1)$  も求めたい。式 **fci|g** を手で微分するのは大変であるし、**s%4b% @S Gq^zS zB^** の結果も長大である。しかし自動微分ならばかなりすっきりしたコードになる。

まず関数 **fciccg** に存在する演算は可算と乗算であるので、式 **fciJg fci|g** に対応した双対数の可乗算を定義する。

1) 多項式環の商環のことをしめしている。詳しくはたとえば雪江明彦「環と体とガロア理論」などを参照。

```

c @CH \~Yf †> %g=
|   -> 4 E †
{   <> @ E %o
j   qCz~q^ - 1<> - 1@j 4l<
I
v @CH - @af †> %g=
u   -> 4 E †
D   <> @ E %o
-   qCz~q^ - j <> 4 j @

```

ここに  $x, y$  はそれぞれ双対数である。次に関数 `fciccg` に存在する要素関数は  $s^S x$ ,  $\langle bs x$ ,  $YbLx$  である。式 `fcidg` を実装するには、双対数の実部と非実部をペア  $(a, b)$  の形式であらわすと、関数  $F(x)$  に対し

$$(x, dx) \rightarrow (F(x), F'(x)dx) \quad \text{fcic\{g}$$

となるように入出力を与えればよいことがわかる。そこで

```

c S\ebqz ^~\e%o s ^e
| @CH <bsf †g=
{   -> 4 E †
j   qCz~q^ ^ei <bsf-g> Q ^ei sS^f-g14
I
v @CH sS^f †g=
u   -> 4 E †
D   qCz~q^ ^ei sS^f-g> ^ei <bsf-g14
-
cE @CH YbLf †g=
cc  -> 4 E †
cl  qCz~q^ ^ei YbLf-g> 4w

```

これで終わりである。あとは

```

c H E Y \4@ †= - @af YbLf \~Yf <bsf †g> sS^f †ggg> sS^f †gg
| @H E Y \4@ †= HF9†>ci EG
{ @H ci EG `QNfQE{ | JCEvDcI | u CEv> QE{ul CE|DC†DI | J{cJJg

```

とすれば  $(F(1), F'(1))$  が求まる。 $G'(x)$  も簡単で、

```

c L E Y \4@ †= HFH †gg
| @L E Y \4@ †= Lf9†>ci EG
{ @L ci EG `QNfQi DDcvCEvu | IuvD_uu> Qii { _cIII CEJJvcJDI g

```

ddX	4- <V@
] ~\ d%ϕ	T, †
yC^sbq b...dq4 4S%o	yC^sbq b...
d%ϕ	d%bqP
d%o;	, C- q wT, †
3Y<W, † *	T, †

表1.1 代表的な PPL とそのバックエンド。\*: BlackJAX は厳密には PPL ではなくサンプラーである。さまざまな PPL 上で利用可能。

とするだけである。以上の例からわかるように自動微分の実際の計算には代数計算を用いているため数値微分に比べて誤差の蓄積は少ない。またフレキシブルにコーディングが可能である。

## ci|

確率プログラミング言語 `fdq4 Ysz dϕLq \ \ S L X ^L~ LQ ddXg` は確率モデルを扱うためのプログラミング言語のことである。様々なものが存在するが、ほとんどのものはバックエンドとして自動微分パッケージを用いている。表 `ci` に `d%Pb^` で動く代表的な `ddX` とそのバックエンドを示す。

本文書では `T, †` をバックエンドとする `] ~\ d%ϕ` を用いる。

|

# T, †

T, † [9](#): は `zC'sbqP b..` や `d%dbqP` 等と同様に自動微分 [9](#)> |: を利用することができる。またその構文は `^~\ e%` と形式的にほぼ対応している。また、後の章では T, † を用いた最適化や [ ; [ ; などを行う。そのためまず T, † で様々な関数を作成できるようにしよう。本章ではまず T, † で一通り自動微分を使用してみたから、第 [1J](#) 章の自動微分のカスタマイズの説明の際に、自動微分の原理の説明を行う構成である。逆に言うと自分で自動微分を定義する必要に迫られない限り原理の理解は必要ないところが T, † の素晴らしいところでもあるといえるかもしれない。

## | ic

T, † の自動微分で遊んでみよう。

## | icic

$$f(x) = |x| + s^S x \quad f|icg$$

を自動微分してみよう。まず `U†i^~\ e%` をインポートする <sup>1c</sup>。

```
S\ebqz U†i^~\ e% s U†e
```

`U†i^~\ e%` の関数を利用して

```
@CH H†g=
qCz~q^ U†ei - 4sf†g j U†ei sS^f†g
```

もしくはラムダ関数を利用して

```
H E Y~\4@ †=U†ei - 4sf†g j U†ei sS^f†g
```

<sup>1c</sup> `U†i^~\ e%` は `^~\ e%` と併用することが多く、`U†i^~\ e%` `^e` で `^~\ e%` `b^e` とする流派、`U†i^~\ e%` `U†e` で `^~\ e%` `^e` とする流派がいるようである。本稿では後者を採用する。

のように関数  $f(x)$  を定義する。

自動微分は `Lq @` を用いる。

```
c Hjb\ U † S\ebqz Lq @
```

微分を

```
c @HLq @Hg
```

とすると、例えば、 $f'(1.0)$  は

```
c @Hci g ` → ?CfS<G qq %ci IJQEF{> @%CEHb-z{|g
```

のようにして求まる<sup>1)</sup>。二階微分も同様に

```
c @HLq @Hg
l @@Hci G ` → ?CfS<G qq %QEDJcJuc> @%CEHb-z{|g
```

のように求まる。

上では微分の引数として、スカラーを代入していた。以下のようなベクトルを代入するとエラーとなる。

```
c †fEU^ei YS^se-<Cf Qc Ec Ec Cg
l @H†fg ` → F
```

これには `U†f^-e` を用いてマッピングをすることで可能となる。

```
c Hjb\ U † S\ebqz f^-e
l @HfE^-efLq @Hgg
{ @Hff†-qqg ` → U^ei-qq %o
```

これで図が描けるようになった。

```
c S\ebqz \-zeYbzYS4i e%Ybz -s eYz
l eYzi eYbz f†-qq>Hf†-qqg>ni n>Y-4CYEnAH†gE6†Gj 5sS^7†8Ang
{ eYzi eYbz f†-qq>@Hff†-qqg>ni n>Y-4CYEnHf†gng
j eYzi †Y-4CYf n†ng
l eYzi YLC^@fg
```

<sup>1)</sup> `@Hcg`だとエラーがでる。これは引数に `Sz` を取れないためである。



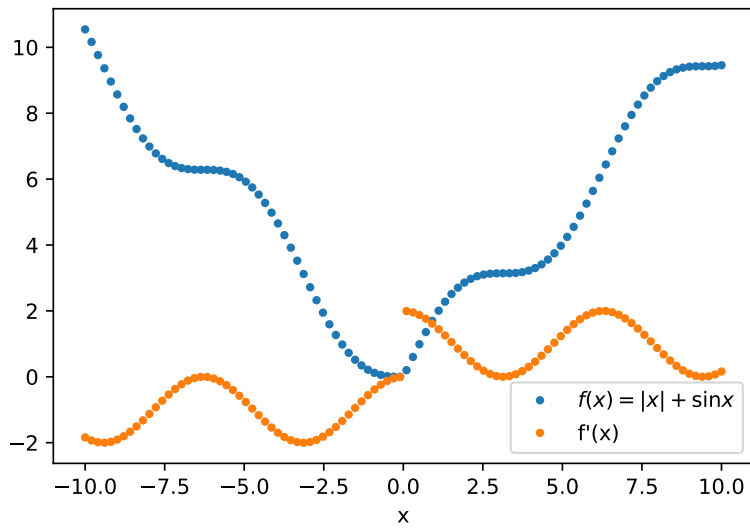


図2.1

$| \mathbf{c}_i |$

次に  $\mathbf{r} = (x, y, z)^\top$  を変数とした多変数関数

$$g(\mathbf{r}) = \sqrt{x^2 + 2y^2 + 3z^2} \quad f[i]g$$

の自動微分を行おう。関数定義は

```

L E Y \4@ †>%x = U^ei sl qzf †11|j | 1%bl|j { 1< 11|g

```

である。偏微分

$$\partial_x g(\mathbf{r}) = \frac{\partial g(\mathbf{r})}{\partial x} \quad f[i]g$$

として  $\mathbf{r}_1 = (1, 1, 1)^\top$  での  $\partial_x g(\mathbf{r}_1)$  は

```

@ @ # ELq @ L > qL ^ ~ \ s E C g
@ @ # f ci > ci > ci g ` -> ? C f S < C, qq % % C E J C E | J D | D > @ % @ C E H b - z { | g

```

ここで  $\mathbf{q}^{\wedge} \mathbf{s}$  が微分する変数を指定している。したがって

$$\partial_y g(\mathbf{r}) = \frac{\partial g(\mathbf{r})}{\partial y} \quad f[i]g$$

として  $\mathbf{r}_1 = (1, 1, 1)^\top$  での  $\partial_y g(\mathbf{r}_1)$  は

```

c @L@%Lq- @L>- qL^~\sEcg
| @L@%fci >ci >ci g ` → ?CfS<C qq- %fEDcvJ_vII> @%CEHb- z{ |g

```

となる。

もしくは

$$\nabla g(\mathbf{r}) = \frac{\partial g(\mathbf{r})}{\partial \mathbf{r}} \quad \text{f|i|g}$$

は、

```

c @L@qELq- @L>- qL^~\sEfEc>|gg
| @L@qfci >ci >ci g
{ ` → f?CfS<C qq- %fEJDE|JD|D> @%CEHb- z{ |g>
J ` ?CfS<C qq- %fEDcvJ_vII> @%CEHb- z{ |g>
| ` ?CfS<C qq- %fci ||JwJD> @%CEHb- z{ |gg

```

となる。

さて、多変数の場合も、バッチ入力する際、そのまま入力するとエラーとなってしまふ。

```

c ‡fEU^ei YS^se- <Cf QcEcEcCEg
| @L@qf ‡- qq>ci >ci g ` →

```

そこでやはり f -e を利用する。この場合、どの変数をマッピングするか指定する。

```

c @L@qfEf^- efLq- @L>- qL^~\sEfEc>|gg> fEc ]b^C>|b^Cg> Ec

```

fEc] b^C] b^Cgの部分で指定している。

```

c @@EL@qff ‡f>ci >ci g
| eYzi eYbzf ‡f>@@@E>Y- 4CYEnA5e- qzS- Y€ ‡ Lf ‡>c>cgAng
{ eYzi eYbzf ‡f>@@@c: >Y- 4CYEnA5e- qzS- Y€%Lf ‡>c>cgAng
J eYzi eYbzf ‡f>@@@|: >Y- 4CYEnA5e- qzS- Y€ ‡ Lf ‡>c>cgAng
| eYzi ‡fY- 4CYf n‡ng
v eYzi YCLC^@f g

```

で、図|i|を描くことができる。

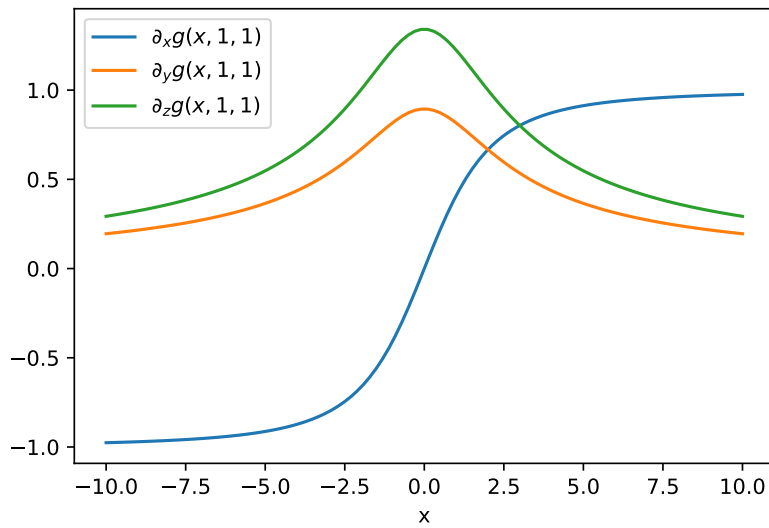


図2.2

入力

入力がベクトルの場合も考えよう。例えばとても簡単な

$$f(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$$

の場合

の場合を考えよう。

```

> %f = dot(x, y)
> %dfdx = y
> %dfdy = x
  
```

この場合でも入力がスカラーだったときと同様に自動微分を計算できる。例えば  $\partial_x f(\mathbf{x}, \mathbf{y})$  は

```

> %f = dot(x, y)
> %dfdx = y
> %dfdy = x
> %dfdx = y
> %dfdy = x
  
```

のように計算できる。

$f$  を用いれば複数の入力ベクトルに対し、マッピングできる。例えば、1 つの  $\mathbf{x}$  を  $f$  で同時に処理してみよう。

```

> %f = dot(x, y)
> %dfdx = y
> %dfdy = x
  
```

ここでも  $f(x)$  の  $f(x)$  に注意しよう。1 つの  $f(x)$  に対して同時に処理するには

```

c | EU^ei YS^se- <Cf C>|I gi qCsP- eCfI >ccg
l | ^EU^ei YS^se- <Cf c>|>I gi qCsP- eCfI >ccg
{ | f^- efLq- @H- qL^- \sEG>f CEG>Cf†>^g ` → ?CfS<C qq- %99ci iig

```

のように、 $f(x)$  の引数を  $f(x)$  にすれば良いこともわかる。

### iciJ

経済学で用いられる  $u(x)$  を考えよう。 $u(x)$  は上に凸 ( $u''(x) < 0$ ) であるとして、 $u'(x) = p$ ,  $u''(x) = pp$ ,  $u'''(x) = ppp$ ,  $u^{(4)}(x) = pppp$

$$p, (x) \equiv -\frac{u''(x)}{u'(x)} \quad f|iug$$

$$pp, (x) \equiv -\frac{u'''(x)}{u''(x)} \quad f|iDg$$

をプロットしてみよう。以下では、例として  $u(x) = \ln(1+x)$  としている。

```

c | `s- \eYC ~zSYsz%aH^<zSb^
l | @CH ~f†g=
{ | qCz~q^ U^ei YbLf ci C†g

```

$f(x)$  を用いてマッピングした後に演算することで  $u(x)$  の形式を引数として受け付けるようになる。

```

c | @CH, p f†- qqg=
l | @^Ef^- efLq- @Lq- @ ~ggg
{ | 4^-Ef^- efLq- @ ~gg
j | qCz~q^ @C^f†- qqg4^-f†- qqg

```

```

c | @CH pp f†- qqg=
l | @^Ef^- efLq- @Lq- @ ~ggg
{ | 4^-Ef^- efLq- @ ~gg
j | qCz~q^ @C^f†- qqg4^-f†- qqg1†- qq

```

プロットは

```

c | †EU^ei YS^se- <Cf C>|>ccg
l | eYzi eYbzf †€>~f†€g>Y- 4CYEnA-f†gAng
{ | eYzi eYbzf †€>, p, f†€g
j | >Y- 4CYEn, p, Af†g 5C ~Sf Q ~/75eqS\C5eqS\C8f†gw-/5eqS\C f†gAn>YsEn@ sPCang

```

```

i eYzi eYbzf †€>pp, f †€g
v >Y- 4CYEnpp, Af †g 5Cl ~Sf Q ~/75eqS\C5eqS\C8f †gw-/5eqS\C f †g †An>YsEn@bzzC@ng
u eYzi YCLC^@f g
D eYzi †Y- 4CYf nA†Ang
- eYzi †YS\ f †E>{i †G
cæ eYzi %MS\ f †E{ g

```

のようにすればよい。図|f|に結果を示す。

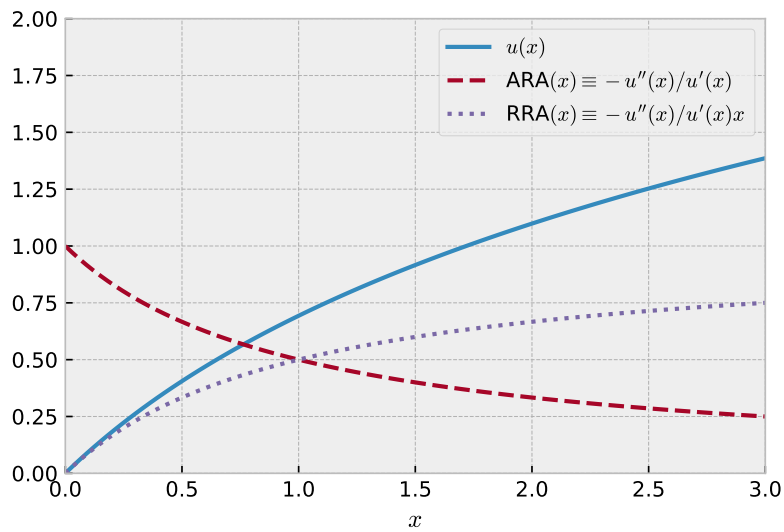


図2.3

$$u(x, A) = \mathbf{Y}\mathbf{L}(A + x) \quad \text{f|i_g}$$

のようにコントロールパラメタ A を入れた場合についても例示しよう。この場合、

```

c @CH ~f ††>, g=
i qCz~q^ Uei YbLf, j †g

```

を用いることになる。さてここでは更に、p, や pp, の微分を図示しよう。すなわち

$$\partial_x p, (x) = \frac{\partial}{\partial x} - \frac{u''(x)}{u'(x)} \quad \text{f|icG}$$

$$\partial_x pp, (x) \equiv \frac{\partial}{\partial x} - \frac{u''(x)}{u'(x)}x \quad \text{f|iccg}$$

をさまざまな  $A$  で図示したい。これには微分があくまで第  $\mathbb{C} \mathbb{d} \setminus \mathbb{C}^z$  について行われていることに注意し、 $f \setminus e$  は最後にこれまた第  $\mathbb{C} \mathbb{d} \setminus \mathbb{C}^z$  についてマッピングすることに注意すると

```

c @CH @ p f †- qq>, g=
l @^ELq_ @fLq_ @f~>- qL^~\sEf Cgg>- qL^~\sEf Cgg
{ 4~^ELq_ @f~>- qL^~\sEf Cgg
J LEY \4@ †>, =Q@C^f †>, gw4~^f †>, g
I PEf^- efLq_ @L>- qL^~\sEf Cgg> f C |b^Cg> Cg
v qCz~q^ Pf †- qq>, g

```

および

```

c @CH @pp, f †- qq>, g=
l @^ELq_ @fLq_ @f~>- qL^~\sEf Cgg>- qL^~\sEf Cgg
{ 4~^ELq_ @f~>- qL^~\sEf Cgg
J LEY \4@ †>, =Q@C^f †>, gw4~^f †>, g †
I PEf^- efLq_ @L>- qL^~\sEf Cgg> f C |b^Cg> Cg
v qCz~q^ Pf †- qq>, g

```

のように定義すれば良いことがわかる。図 |ij は

```

c , - qqE9Qi CCECci C
l HSL EeYzi HSL~qCf HSLsS< Cef cl >{ gg
{ Hbq S>, S^ S^ C^~\Cq_ zCf, - qqg=
J - †EHSLi - @@s~4eYbzf c>{ >Sj cg
I †EU^ei YS^se- <Cf Q S^j CEI >Q S^j { >cCEg
v eYzi zSzYcf n, Enj szqf, S^gg
u - †i eYbzf †€~f †€>, S^g>Y- 4CYEnA-f †gAng
D - †i eYbzf †€>@ p, f †€>, S^g>Y- 4CYEnA5e- qzS- Y€†A p, Af †gAn>YsEn@ sPC@ng
_ - †i eYbzf †€>@pp, f †€>, S^g>Y- 4CYEnA5e- qzS- Y€†App, Af †gAn>YsEn@bzzC@ng
cE eYzi YCLC^@f g
cc eYzi †Y- 4CYf nA†Ang

```

のように書ける。

## |i| Hbq

さてこれまで比較的簡単な関数を自動微分してきた。この程度なら手で（もしくは  $s\% e\%$  とか  $\setminus -zPC \setminus -zS<$  で）微分できるじゃないかといわれそうである。まあ別に究極的には手でできるのだが、いかにも面倒くさそうな場合として関数が  $Hbq Yb e$  を含む場合<sup>14</sup>、自動微分を使えると便利そうである。

<sup>14</sup> 古代の言語  $Hbqzq^{\wedge}$  ならば  $@b$  ループのことである。

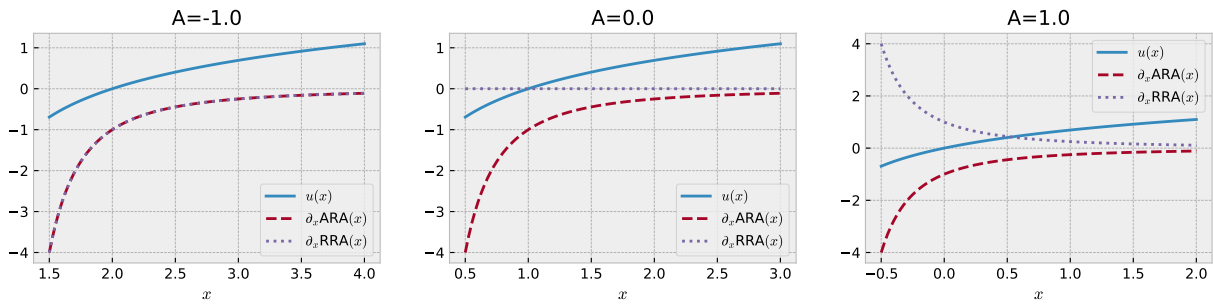


図2.4

U†ではHbqYbというのがあるが、fCsS^Civの時点ではこれを用いると自動微分が通らない。そこで用いるのがU†Y†is<^である。s<^と等しいe%Pb^コードはU†公式チュートリアルによると

```

c @CH s< ^fH> S^Sz> †s> YC^LzPE] b^Cg=
l SH †s Ss ] b^C=
{ †s E 9] b^C: 1 YC^LzP
J <- qq%oE S^Sz
I %E 9:
v Hbq † S^ †s=
u <- qq%o %E H<- qq%o †g
d %si - eeC^@f%g
- qCz~q^ <- qq%o ^ei sz-<W%sg

```

である。これはHbqYbより一般的だが、HbqYbにしようと思うとなんかちよつと頭がくらくらする。†sを] b^Cとして、YC^LzPをHbqYbの回数として、ほぼ同じコードなのだが以下のように書き換える。

```

c @CH s< ^fH> †CE YC^LzPE^g=
l szCe E 9] b^C: 1 ^
{ † E †CE
J ^- qq E 9:
I Hbq S S^ szCe=
v †> ^~YY E H†> Sg
u ^- qqj - eeC^@f^~YYg
d qCz~q^ †> ^ei sz-<W^ qqg

```

これは、特に使っていない^~YYとか^-qqを無視すれば、ほとんど以下のHbqYbと同じである

```

c @CH s< ^fH> †CE^g
l †E†CE
{ Hbq S S^ q^LCfCE^g=
J † E H†g

```

```
1 qCz~q^ †
```

というわけで、`s<-^`で `Hbq Ybce` が実装できることが分かった。

さて、例として

$$f(x) = 1/(1+x) \quad \text{f|ic|g}$$

を  $n$  回適用させた関数 `Lf†t^g` を考えよう。例えば  $n = 3$  の時、

$$g(x; 3) = f(f(f(x))) = \frac{1}{1 + \frac{1}{1+x}} \quad \text{f|ic\{g}$$

である。これを微分した関数

$$h(x; n) = \frac{\partial}{\partial x} g(x; n) \quad \text{f|icJg}$$

を計算する。

まず `s<-^>Lq@` を `S\ebqz` する。

```
c Hb\ U † Y † S\ebqz s<- ^
1 Hb\ U † S\ebqz Lq@
```

関数  $f$  を定義する。

```
c @CH H†>^~YYg=
1 †Eci Cf ci Cf†g
† qCz~q^ †>^~YY
```

`^~YY` は別に意味のない変数だが `s<-^` の形式に合わせるために無駄についている。次に `s<-^` で `Hbq Ybce` 部分を実装する。今は  $n = 3$  で試そう。

```
c @CH Lf†Cg=
1 †>^~YYEs<- ^fH†C] b^C>\{g
† qCz~q^ †
```

そして自動微分を定義する。ここでベクトル化した `†-qq` を代入するために `f^-e` しておこう。`USz` というのは `T-szq\y S C` コンパイラ (実行時コンパイラ) のことで、コンパイルするので高速になる。今回の例では `USz` をするのとしないので雲泥の差が出るので試されたし。

```
c Hb\ U † S\ebqz f^- e
1 Hb\ U † S\ebqz USz
† PEUSzf f^- efLq @Lggg
J †- qqEU^ei YS^se- <Cf Cc>cCEg
1 P- qqEPf †- qqg
```



さて、答え合わせを行うために  $s\%e\%a$  も計算しておこう。

```

c S\ebqz s%e%o s se
| † E sei r%4bYf o†og
{ @CH Hsef †g=
J qZ~q^ cwf cj †g
I sei @SHH Hsef Hsef Hsef †ggg>†g

```

答えが、

$$h(x, 3) = -\frac{1}{1 + \frac{1}{1 + \frac{1}{x+1}} \cdot \frac{1}{1 + \frac{1}{x+1}} (x+1)^2} \quad f|i|c|g$$

と求まる。つまり比較すべき関数は

```

c PseEY \4@ †=Q Hsef Hsef Hsef †ggg|1| 1f Hsef Hsef †ggg|1| 1f Hsef †gg|1|

```

となる。比較の図は、

```

c S\ebqz \-zeYbzYS4i e%Ybz -s eYz
| S\ebqz sC 4bq^ -s s^s
{ eYzi sz%CI ~sCf o4\ Pog
J
I HSL EeYzi HSL~qCf g
v - †EHSLi - @@Es ~4eYbz f | ccg
u eYzi eYbz f †- qq>Psef †- qqgQ- qqg
D eYzi %& 4CYf ns%e%QJ †ng
- - †EHSLi - @@Es ~4eYbz f | c | g
cE eYzi eYbz f †- qq>P- qqg
cc eYzi †Y 4CYf nA†Ang
cl eYzi %& 4CYf nA5e- qzS- YEt HHHH †gggAng
cE eYzi sPb. fg

```

でかけ、結果は、図 [ii](#) である。このように直接関数定義した場合と  $v\mathbf{u}$  桁くらいの精度で一致しているのがわかる。

[ii](#) |ic

$s\< \wedge$

$U\ddagger iY \ddagger is\< \wedge$  の他の使い方を練習してみよう。例えば  $\mathbf{x} = (x_0, x_1, x_2, x_3)^T$  に対して関数

$$f(\mathbf{x}) = x_0 + \frac{1}{x_0} \quad x_1 + \frac{1}{x_1} \quad x_2 + \frac{1}{x_2} \quad x_3 + \frac{1}{x_3} \quad f|i|c|v|g$$

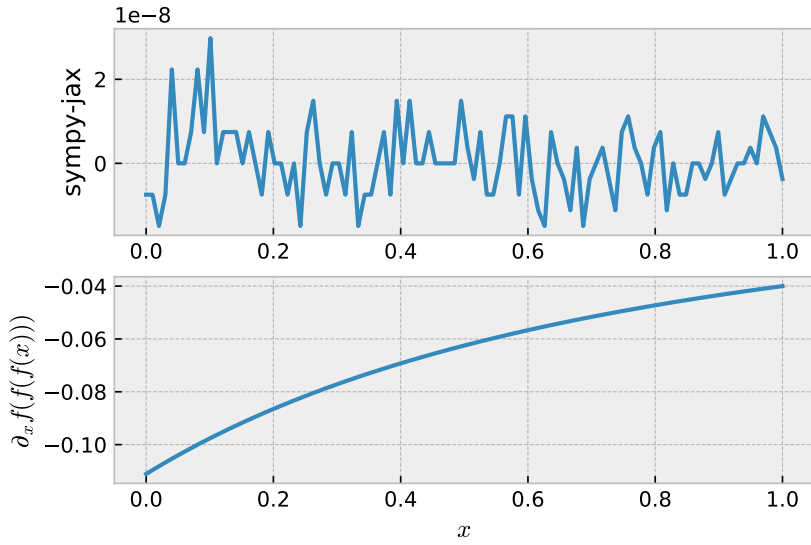


図2.5

を定義する。この関数の微分

$$\mathbf{h}(\mathbf{x}) \equiv \nabla f(\mathbf{x}) = \left( \frac{\partial f(\mathbf{x})}{\partial x_0}, \frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \frac{\partial f(\mathbf{x})}{\partial x_3} \right)^\top$$

の自動微分を実装してみよう。式  $y_i = y_{i-1}x_i + 1/x_i$  を  $\mathbf{x}$  に用いればよいことが予想される。式  $f(\mathbf{x})$  は内側のかっこから再帰的に  $(\cdot)x_i + 1/x_i$  を計算していけばよいことに気づく。そこで、まず

```

c @CH Lf %t g=
l   %t %t j ci %t
{   qz~q^ %t b^C

```

とする。そして  $\mathbf{s}^{\wedge}$  を用いて

```

c 2LSz
l @CH H tsg=
{   %t ci %t
j   %t ~YEs< ^fL> %t tsg
l   qz~q^ %t

```

とすれば再帰的に  $y_i = y_{i-1}x_i + 1/x_i$  を計算できることがわかる。この自動微分は単純に

```

c PELq @Hg

```

とすればよい。これで、例えば、

```

c | $sEU^ei - qq- %9ci >|i>Ji>Ii: g
| Pf$sg ` → ?CfS<C qq- %9CE > {Ii > ||i cDiI> cD|c :> @%CEHb- z{|g

```

と求まる。この形式はいくらでも要素数を増やせる利点がある。すなわち例えば、

```

c | PfU^ei - qq- %9ci >|i>{i>Ji>Ii>vi>ui>Di: gg
| ` → ?CfS<C qq- %9CE> {I|DCE> |_J_{i{|> |{c{Ii
{ ` > cDv/iII-> cIIu/i_u > c{I CEuuCE> ccvD|i c_J :> @%CEHb- z{|g

```

といった感じである。

## |i{ TRy

クラスを定義してクラス内の中においた関数に TRy をおこなえると便利である。この場合は、以下のように  $H^{\langle zbbYs \rangle} e-qzS-Y$  を用いてデコレータをつけるとよい。

```

c | Hjb\ H^<zbbYs S\ebqz e- qzS- Y
| <Y-ss HkU $fb4UC<zg=
{ @CH €€S^Sz€€f sCYH<>^g=
J sCYH <E<
I sCYH ^E^
v
u @CH H sCYH $>^~Yg=
D $EsCYH <v f sCYH <j $g
- qCz~q^ $>^~YY
cCE
cc 2e- qzS- YfUSz> sz- zS<€ qL^~\sEf Cgg
cl @CH Lf sCYH $&Cg=
cf $>^~YEs<- ^fH $&C| b^C>sCYH ^g
cJ qCz~q^ $
cl
cv 2e- qzS- YfUSz> sz- zS<€ qL^~\sEf Cgg
cu @CH Pf sCYH $&Cg=
cD PPEf\ - efLq- @f sCYH Lgg
c_ qCz~q^ PPF $&Cg

```

このようにクラスで書くことにより、例えば、関数の中を引数を変えて

```

c | Hbq S S^ q- ^LCf c>Jg=
| Hk<EHkU $fci CgSg

```

```

{ eYzi eYbzfHk<i Pf †- qqg>Y- 4CYEn^Enj szqf Sgg
J eYzi YCLC^@f g
I eYzi sPb.f g

```

のように使うことができる。また、TRY で実行時にコンパイルされてしまうので、一度、LやPを呼び出して  
 しまうと、その後に Hk<i^EJ のように更新しても、関数の方の Hk<i^ は変更されないことに注意が必要である

```

c Hk<EHkU †f ci Ccg
I Hk<i ^E|
{ Hk<i Pf †- qqg ` ^E|
J Hk<i ^E|
I Hk<i Pf †- qqg ` ^E|

```

## |iJ

ここまでは U†i^~\ e%で定義された関数を組み合わせた自動微分を利用してきた。U† は自分で関数の微分  
 を定義できる。ここでは  $\mathbf{x} \in \mathbb{R}^N, \mathbf{f} \in \mathbb{R}^M$  として、関数  $\mathbf{f}(\mathbf{x})$  を考える。自動微分は、T- <b4S ^、

$$J_{f;x} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{matrix} \text{O} & & & & 1 \\ \text{B} & \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \text{B} & \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \text{B} & \vdots & \vdots & \ddots & \vdots \\ \text{B} & \frac{\partial f_M}{\partial x_1} & \frac{\partial f_M}{\partial x_2} & \cdots & \frac{\partial f_M}{\partial x_N} \end{matrix} \in \mathbb{R}^{M \times N} \quad f|icDg$$

を計算するのが目的であるが、Gdq .: q@\ b@C と pCfCpC\ b@Cf4- <Wepd- L- zB^ ともgという二つのやり  
 方がある。これは微分の連鎖則をどちらから解いていくかという違いである。まず  $\mathbf{f}(\mathbf{x}) = \mathbf{f}_2(\mathbf{f}_1(\mathbf{x}))$  という  
 場合の連鎖がどうなるか考えよう。 $\mathbf{f}$  の微分は  $\mathbf{x}_1 = \mathbf{f}_1$  と書くと、

$$J_{f;x} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \frac{\partial \mathbf{f}_2(\mathbf{x}_1)}{\partial \mathbf{x}_1} \frac{\partial \mathbf{x}_1}{\partial \mathbf{x}} = \frac{\partial \mathbf{f}_2(\mathbf{x}_1)}{\partial \mathbf{x}_1} \frac{\partial \mathbf{f}_1(\mathbf{x})}{\partial \mathbf{x}} \quad f|ic_g$$

$$= J_{f_2;x_1} J_{f_1;x} \quad f|i|Cg$$

である。つまり一般に  $\mathbf{f}(\mathbf{x}) = \mathbf{f}_L(\mathbf{f}_{L-1}(\cdots \mathbf{f}_2(\mathbf{f}_1(\mathbf{x})))$  の場合、

$$J_{f;x} = J_{f_L;x_{L-1}} \cdots J_{f_2;x_1} J_{f_1;x} \quad f|i|cg$$

ということである。

## |iJic T- <b4S ^, Gzbq dφ@<z

まず以下のような T- <b4S ^, Gzbq dφ@<z fT, dg

$$T, d(\mathbf{f}, \mathbf{u}) = J_{f;x} \mathbf{u} = \frac{\partial f_1}{\partial \mathbf{x}} \mathbf{u}, \dots, \frac{\partial f_M}{\partial \mathbf{x}} \mathbf{u}^T \quad f|i||g$$

が用意できれば、 $\frac{\partial f_i}{\partial x_j}$  は、

$$T, d(\mathbf{f}, \mathbf{e}_j) = \frac{\partial \mathbf{f}}{\partial x_j} \quad \mathbf{f} | \mathbf{i} | \mathbf{g}$$

の  $i$  成分を取り出すことで計算できる。ここに  $\mathbf{e}_j$  は  $j$  番目の単位ベクトル ( $j$  番目の要素だけが  $\mathbf{c}$  で残りが  $\mathbb{C}$  のベクトル) である。式  $\mathbf{f} | \mathbf{i} | \mathbf{c} \mathbf{g}$  の右辺に  $\mathbf{u}$  をかければ

$$\begin{aligned} J_{f_i; \mathbf{x}} \mathbf{u} &= J_{f_L; x_L} \cdots J_{f_2; x_1} J_{f_1; \mathbf{x}} \mathbf{u} & \mathbf{f} | \mathbf{i} | \mathbf{J} \mathbf{g} \\ &= J_{f_L; x_L} (\cdots J_{f_3; x_1} (J_{f_2; x_1} (J_{f_1; \mathbf{x}} \mathbf{u})) \cdots) & \mathbf{f} | \mathbf{i} | \mathbf{I} \mathbf{g} \end{aligned}$$

という風に、右 (小さい  $L$  の順) から  $T, d$  を適用していけば  $J_{f_i; \mathbf{x}} \mathbf{u}$  ひいては  $J_{f_i; \mathbf{x}}$  の各成分が求まることがわかる。これを  $\mathbb{H} \mathbf{q} \cdot \cdot \mathbf{q} @ \setminus \mathbf{b} @ \mathbb{C}$  という。つまり  $\mathbb{H} \mathbf{q} \cdot \cdot \mathbf{q} @ \setminus \mathbf{b} @ \mathbb{C}$  に必要なのは  $T, d(\mathbf{f}_1, \mathbf{u}), T, d(\mathbf{f}_2, \mathbf{u}), \dots, T, d(\mathbf{f}_L, \mathbf{u})$  であるということになる。

というわけで  $T, \dagger$  に戻って、 $\mathbb{H} \mathbf{q} \cdot \cdot \mathbf{q} @ \setminus \mathbf{b} @ \mathbb{C}$  のカスタム自動微分は、 $T, d$  を定義する  $\mathbb{U} \dagger \leftarrow \mathbf{szb} \setminus \mathbb{E} \mathbf{U} \mathbf{f} \mathbf{e}$  を使用すれば良い。さて以下の例では

$$f(\mathbf{x}) = x_2 \mathbb{S} x_1 \quad \mathbf{f} | \mathbf{i} | \mathbf{v} \mathbf{g}$$

の  $T, d$  を実装する。みやすさのため  $x = x_1 > y = x_2$  とおいているのに注意。

```

c Hjb\ U † S\ebqz <-szb\€flUe
| 2<-szb\€Ufe
{ @CH H † > %g=
j qCz~q^ Uei sS^f † g 1 %o
I
v 2H @CHUfe
u @CH H€Ufef eqS\ - Ys> z- ^LC^zsg=
D † > %oE eqS\ - Ys
_ ~† > ~%oE z- ^LC^zs
eE @H † E%o1 Uei <bsf † g
cc @H @Uei sS^f † g
eI eqS\ - Y€b~z E H † > %g
eE z- ^LC^z€b~z E @H † 1 ~† j @H @%o1 ~%o
cJ qCz~q^ eqS\ - Y€b~z > z- ^LC^z€b~z

```

上で、 $z- ^LC^z€b~z E @H † * ~† j @H @% * ~%$  とおいているが、これが  $T, d$  になっている。つまり、

$$\begin{aligned} T, d(f, \mathbf{u}) = J_{f; \mathbf{x}} \mathbf{u} &= \frac{\partial f}{\partial \mathbf{x}} \mathbf{u} = \frac{\partial f(x, y)}{\partial x} u_1 + \frac{\partial f(x, y)}{\partial y} u_2 & \mathbf{f} | \mathbf{i} | \mathbf{u} \mathbf{g} \\ &= (y \langle \mathbf{b} \mathbf{s} x \rangle u_x + (\mathbb{S} x) u_y & \mathbf{f} | \mathbf{i} | \mathbf{D} \mathbf{g} \end{aligned}$$

となっているからである。

$\mathbf{J}_i$  ,  $\mathbf{C}^T \mathbf{b}^T \mathbf{A}^T \mathbf{d} \mathbf{p}$

T, d では右から  $\mathbf{u}$  をかけて  $\mathbf{T} \mathbf{b}^T \mathbf{A}^T$  の成分を取り出したが、左から書けるのが、  $\mathbf{C}^T \mathbf{b}^T \mathbf{A}^T \mathbf{d} \mathbf{p}$  である。つまり

$$\mathbf{Td}(\mathbf{f}, \mathbf{u}) = \mathbf{u}^T \mathbf{J} = \mathbf{u} \cdot \frac{\partial \mathbf{f}}{\partial x_1}, \mathbf{u} \cdot \frac{\partial \mathbf{f}}{\partial x_2}, \dots, \mathbf{u} \cdot \frac{\partial \mathbf{f}}{\partial x_N} \quad \text{f|i|_g}$$

と定義される。この場合も同様に  $\mathbf{u} = \mathbf{e}_i$  とおけば、

$$\mathbf{Td}(\mathbf{f}, \mathbf{e}_i) = \mathbf{e}_i^T \mathbf{J}_{f;x} = \frac{\partial f_i}{\partial x_1}, \frac{\partial f_i}{\partial x_2}, \dots, \frac{\partial f_i}{\partial x_N} = \frac{\partial f_i}{\partial \mathbf{x}} = (\nabla f_i)^T \quad \text{f|i|_g}$$

が得られるので、 $\mathbf{T} \mathbf{b}^T \mathbf{A}^T$  各成分が求まることになる。  $\mathbf{Td}$  を用いる連鎖則の計算法は `qfCpC\b@c` とよばれ、左から

$$\begin{aligned} \mathbf{u}^T \mathbf{J}_{f;x} &= \mathbf{u}^T \mathbf{J}_{f_L;x_{L-1}} \cdots \mathbf{J}_{f_2;x_1} \mathbf{J}_{f_1;x} & \text{f|i|_g} \\ &= (\cdots ((\mathbf{u}^T \mathbf{J}_{f_L;x_{L-1}}) \mathbf{J}_{f_{L-1};x_1}) \mathbf{J}_{f_{L-2};x_1} \cdots) \mathbf{J}_{f_1;x} & \text{f|i|_g} \end{aligned}$$

となり  $4 \times W \times q \times S \times C \times S \times S$  に必要なのは、  $\mathbf{Td}(\mathbf{f}_L, \mathbf{u})$ ,  $\mathbf{Td}(\mathbf{f}_{L-1}, \mathbf{u})$ ,  $\dots$ ,  $\mathbf{Td}(\mathbf{f}_1, \mathbf{u})$  ということになる。

T, † に戻って、 `qfCpC\b@c` のカスタム自動微分は、  $\mathbf{Td}$  を定義する `U†i~\e%` を使用すれば良い

```

c Hjb\ U † S\ebqz <-szb\efUe
|
{ 2<-szb\efUe
j @CH Pf †> %g=
i qCz~q^ U^ei sS^f†g 1 %o
v
u @CH PE4.@ qCs> ~g=
d <bs€†> sS^€†> %oE qCs
_ qCz~q^ f%o1 <bs€† 1 ~> sS^€† 1 ~g
cE
cc @CH PEH.@ †> %g=
cl qCz~q^ Pf †> %g> fU^ei <bsf†g> U^ei sS^f†g> %g
cf
cJ Pi @CHfUef PEH.@ PE4.@g

```

$\mathbf{U}^{\dagger i \sim \epsilon}$

$\mathbf{U}^{\dagger i \sim \epsilon}$  は直接のインデックスアサインはできない。以下のコードでは5成分のゼロアレイを用意して、インデックス `c>` にそれぞれ `cE` を代入しようとしている。しかし、最後の行はエラーになる。

```

c - EU^ei < CqbsfI g
| S^@#EU^ei - qq- %f9c> |: g
{ f- YEU^ei - qq- %f9cCEB{i CEg
j - 9S^@#: Ef- Y `

```

そこで  $e \sim c H^{\langle z b^{\wedge} - Y b e C q z b q s \rangle}$  を格納している `U # i b e s` モジュールから `S^@#` に関するいくつかの関数を用いることができる。代わりに

```

c Hjb\ U # i b e s S\ e b q z S^@#€-e@ zC
| Hjb\ U # i b e s S\ e b q z S^@#€ -s UbS
{ -ES^@#€-e@ zCf- >UbS9S^@#: >f- Yg
j ` ?CfS<G, qq- %f9 CE> cCE> {i> CE> CE: > @%CEHb- z{|g

```

とすればよい。

ここですすである配列のインデックスアサインをして足すには `S^@#€-@@` を用いる。

```

c Hjb\ U # i b e s S\ e b q z S^@#€-@@
| S^@#€-@@f- >UbS9U^ei - qq- %f9CEc: g >U^ei - qq- %f9c> |: gg
{ ` ?CfS<G, qq- %f9 ci> c/i> {i> CE> CE: > @%CEHb- z{|g

```

{

T, †

{ic U†iCteCqS C^z Y

最適化は自動微分の代表的な応用先である。多くの最適化手法はもとを正せば Kq @C^z? Cs<C^z

$$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \gamma \frac{\partial}{\partial \mathbf{r}} f(\mathbf{r}) \quad \text{f\{icg}$$

に似た方法で更新していくからであり、つまり目的関数の微分  $f'(\mathbf{r})$  が必要だからである。つまり自動微分が通るように構成しておけば、自分で微分を計算する必要はない。

{icic T, † bezS SCq

T, † には U†iCteCqS C^z YbezS SCq という最適化モジュールがある。有名どころの bezS SCq はこのものを使って手軽に最適化できる。例として 3bbzP H^<zSb^

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2 \quad \text{f\{i|g}$$

を目的関数として、これを最小化する  $(x, y)$  を求めてみよう。

```

c `3bbzP H^<zSb^
| @CH 4bbzPf †>%g=
{ Hef †j |i 1%Qi g11| j f|i 1†j %Qi g11|
J qCz~q^ H

```

```

c `
| @CH b4UC<zSfCf qg=
{ HE4bbzPf q9CE>q9c: g
J qCz~q^ H

```

U†iCteCqS C^z YbezS SCq は bez€SSz f初期化g bez€~e@ zC (アップデート) >LC€e-q \s (パラメタをゲットする) の3セットを用いて最適化を行う。ステップを以下のように定義する。

```

c Hjb\ U † S\vebqz f- Y~C€ ^@€Lq @
| @CH szCefz> bez€sz- zCg=

```

|{



```

f- Y-C> Lq @s E f- Y-Ce ^@ELq @f b4UC<zSfCgf LCz€e- q \sf bez€sz- zCgg
bez€sz- zC E bez€-e@ zCfz> Lq @s> bez€sz- zCg
qCz~q^ f- Y-C> bez€sz- zC

```

$f$   $Y-Ce ^@ELq @f$  は引数  $x = p$  での関数  $f(x)$  の値と微分値  $f'(x)$  を返す。  $z$  は  $szCe S^@C$  (整数値) である。微分値と  $szCe S^@C$ 、前回の状態  $fbez€sz- zCg$  を  $bez€-e@ zC$  に渡すことで、次のステップにアップデートする。初期値を  $qE f†@Cg$  として、  $] szCe$  回ステップを回して、途中結果を保存するには

```

@CH @bbezf qEbez€S^Sz>LCz€e- q \s>| szCeg=
bez€sz- zC E bez€S^Szf qCg
†zq- UE9q@CE:
%aq- UE9q@C: :

Hbq z S^ q ^LCf] szCeg=
f- Y-C> bez€sz- zC E szCefz> bez€sz- zCg
eELCz€e- q \sf bez€sz- zCg
†zq- U - eeC^@fe9CEg
%aq- U - eeC^@fe9C: g
qCz~q^ †zq- U> %aq- U> e

```

のようにすればよい。  $bezS S Cq$  としては、  $?$ 、  $[$  と  $rK?$  を試す。

```

Hjb\ U †i C†eCqS\C^z- Y S\ebqz bezS\S< Cqs
\, ?, [
bez€S^Sz> bez€-e@ zC> LCz€e- q \s E bezS\S< Cqsi - @ \f cCQg
qEE U^ei - qq- %f@ui I>Qui I: g `
†zq- Uc> %aq- Uc> eE@bbezf qEbez€S^Sz>bez€-e@ zC>LCz€e- q \s>cCQg

```

もしくは

```

`rK?
bez€S^Sz> bez€-e@ zC> LCz€e- q \s E bezS\S< Cqsi sL@cCQg
qEE U^ei - qq- %f@ui I>Qui I: g
†zq- U|> %aq- U|> eE@bbezf qEbez€S^Sz>LCz€e- q \s>cCQg

```

のような感じである。実行結果を表示しよう。

```

S\ebqz \- zeYbzYS4i e%Ybz -s eYz
S\ebqz U †i ^~\e%o s U^e
S\ebqz sC 4bq^ -s s^s
eYzi sz%Ci ~sCf o4\Pog

```

```

I
v `LqS@Hbq <b^zb-q
u | †Ec[ ] %c[ ]
D †EU^ei YS^se- <Cf Q >cI >] †g
- %EU^ei YS^se- <Cf Q EcE] %g
cE LE4bbzPf †9=>] b^C 1U^ei b^Csff] †>] %gg>% b^C>=: 1U^ei b^Csff] †>] %gg
cc
cI HSL EeYzi HSL~qCf HSLsS< Cef v>vgg
cF eYzi eYbzf †zq- Uc>%zq- Uc>Y.ECEul ><En; cn>Y- 4CYEn, ?, [ ng
cJ eYzi eYbzf †zq- U] >%zq- U] >Y.ECEul ><En; Cb>YsEn@ sPC@n>Y- 4CYEnrK?ng
cI eYzi <b^zb-qf †>%U^ei YbLfLi yg>YCfCYsEcE<\- eEnLq- %a>- YeP- ECE{ g
cv eYzi †Y- 4CYf n†ng
cu eYzi %& 4CYf n%ng
cD eYzi YCLC^@f g

```

結果は、図{icのようになる。これは、?, [ が rK? より悪いというわけではなく szCe€sSCを大きめ

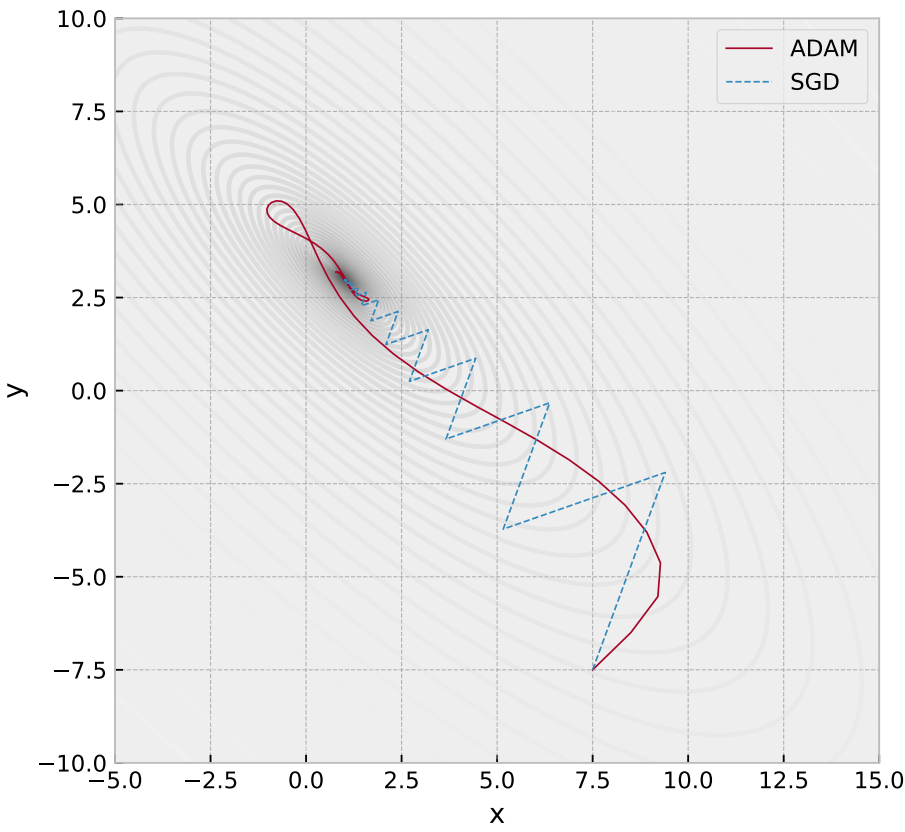


図3.1

に設定したので慣性が効いているのが見えているというわけである。., SMeCS に最適化のテスト関数

`fPzses=wwC^i..S\EC@S ibqLw.S\WycszEH^<zsb^sEhbqf bezS S-zsb^g`が多数挙げられているので試してみよう。全然、大域最小値に収束しないこと請け合いである。

## {i| T, † bez

`T, † bez`は `T, †` を用いた最適化のためのパッケージである。特徴として目を引くのは

```
á ? S Cq^zS 4Y= bezS S-zsb^ ep4Y\ sbY-zsb^s < ^ 4C @S Cq^zS zC@ ...SP qCseCz zb zPCsq S'e-zs
CZPCqS eYSZY%bq fS --zb@S bH^fY@- YbqzP\ S Cq zsb^si
```

すなわち、解自体を最適化問題のパラメタで微分できるというところにある。すばらしい。

以下では  $f(x; a) = (x - s^S(a))^2$  を  $x$  について最適化してみよう。もちろん答えは  $x_* = s^S(a) \equiv g(a)$  となるはずである。

```
c S\ebqz U † bez
| @CH H † > g=
{ qCz~q^ f † Q^ei sS^f- gg11|
j L@ E U † bezi Kq @SC^z?Cs<C^zfH^EH> \- † SzCqEI CEG
```

これだけである。解  $x_* = g(a)$  を以下のように定義すればよい。

```
c @CH Lf- g=
| qCs E L@ q^fS^Sz€e- q \sE^ei q ^@b\i ^bq\ - Yf CECci Cg> - E- g
{ e- q \s> sz- zC E qCs
j qCz~q^ e- q \s
```

これで、

```
c Lf CEG `QN ?CfS<C qq- %CE_D\{Jc> @%CEHYb- z\|g
```

となり解が出る。さらに  $g(a)$  を  $a$  で微分することができる。

```
c Hjb\ U † S\ebqz Lq- @
| @LELq- @Lg
{ @LfQE|g `QN ?CfS<C qq- %CE_DCEEvv> @%CEHYb- z\|g
```

これは  $\frac{\partial}{\partial a} x \setminus s = \frac{\partial}{\partial a} g(a) = \langle bs(a) \rangle$  のことであり、たしかにそうになっている。

## J

# ] ~\ e%ϕ

] ~\ d%ϕ の [ ; [ ; は  $O[ ; Q \} yr f O - \setminus Sz b ^ S ^ [ b ^ z C ; - q \bar{b} ] b \} y \sim q ^ r - \setminus e \bar{S} L g$  が基本である。 $O[ ;$  は勾配情報を用いた高受容率の [ ; [ ; であり、フィットするモデルのパラメタ数  $D$  が多い場合にはランダムメトロポリス・ヘイスティング法より効率がよい。理論的には収束までの時間が  $O[ ;$  では  $D^{5/4}$ 、ランダム [  $O$  では  $D^2$  に比例するとされている  $g$ 。ここでは  $O[ ;$  の説明はしないが、日本語では「ゼロからできるMCMC」fVG 理工学専門書が筆者のおすすめである。 $O[ ;$  では計算に  $XG e \bar{Q} \bar{P} L$  を使い、モデルのパラメタによる微分が必要である。 $\wedge \setminus e \% \bar{\phi} g$  : を用いて  $O[ ; Q \} yr$  をしよう。 $\wedge \setminus e \% \bar{\phi}$  は  $T, \dagger$  を用いた  $d \bar{\phi} 4 - 4 S \bar{S} \bar{z} \times d \bar{\phi} L q \setminus \setminus S L X - ^ L \sim L C$  である。つまり  $T, \dagger$  の自動微分が通るようにモデルを定義しないとらない。

## Jic

正弦曲線から生成されるデータにガウスノイズが加わったものを考えよう。すなわち

$$y = s \bar{S} (x + \phi) + \epsilon \quad f J i c g$$

$$\epsilon \sim \mathcal{N}(0, \sigma) \quad f J i | g$$

のような場合である。ここに  $\epsilon \sim p$  というのは確率変数  $\epsilon$  が確率密度  $p$  から生成されるということを意味している。また  $\mathcal{N}(\mu, \sigma)$  は平均  $\mu$ 、標準偏差  $\sigma$  の正規分布のことである。さてデータから位相  $\phi$  とノイズレベル  $\sigma$  を推定したい。まず模擬データを以下のように生成しておく。

```

c S\ebqz ^~\e%σ s ^e
l ^ei q ^@b\i sCC@f { | g
{ eP- sCECEI
J sSLS^ECE{
l ] E| CE
v ‡E^ei sbqzf ^ei q ^@b\i q ^@f ] gg1J1^ei eS
u %̂^ei sS^f ‡j eP- sCgj ^ei q ^@b\i ^bq\ - Yf ĈsSLS^>sS< CE] g

```

このデータ  $f \ddagger \% g$  は図 [Jic](#) のような感じとなっている。

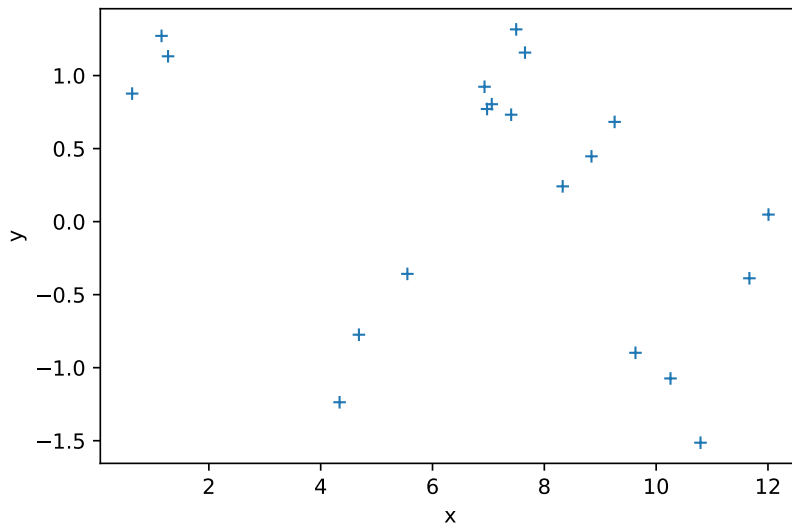


図4.1

さてこのデータに対し確率モデルを設定し  $O[; Q]$  yr で事後分布を求めよう。確率モデルとしては式  $f_{Ji} \{ J_i | g \}$  また事前分布として

$$\begin{aligned} \phi &\sim U[-\pi, \pi] & f_{Ji} \{ g \\ \sigma &\sim B(\beta, \alpha) & f_{Ji} \{ g \end{aligned}$$

としたものを用いる。この確率モデルを  $\hat{\sim} e_{\phi}$  を使って以下のように定義する。

```

c S\ebqz U \phi \sim U[-\pi, \pi]
l S\ebqz \sigma \sim B(\beta, \alpha)
{ S\ebqz y | x, \phi, \sigma \sim N(x + \phi, \sigma^2)
J
r CH \b CY f \{ g =
v eP-sC E \hat{\sim} e_{\phi} s-\eYCF oeP-sCo> @Szi } ^SHbq\ f Qi CEU^ei eS> ci CEU^ei eSgg
u sSL\ - E \hat{\sim} e_{\phi} s-\eYCF osSL\ - o> @Szi B\eb^C^zS- Yf ci gg
D \sim EU^ei sS^f \{ j eP-sCg
- \hat{\sim} e_{\phi} s-\eYCF o\%> @Szi ] bq\ - Yf \sim> sSL\ - g> b4sE\%g

```

この例からわかるように、確率分布は  $@Szi$  で指定されたものを、 $\hat{\sim} e_{\phi}$  の  $s-\eYCF$  内で定義することにより確率変数として扱われる。最後のデータ  $\%$  の部分は式  $f_{Ji} \{ J_i | g \}$  が

$$y \sim \mathcal{N}(s(x + \phi), \sigma) \quad f_{Ji} \{ g$$

と書き直されることよりわかる。さてこれを  $\hat{\sim} e_{\phi}$  の  $O[; Q]$  yr にかける。

```

c H\ b \ U \phi S\ebqz q- ^@ \

```

```

| Hjb\ ^~\e%qbi S^HCq S\ebqz [; [; > ]}yr
{ q^LÉVC%oE q ^@b\i dp] KVC%foCg
J q^LÉVC%o q^LÉVC%6 E q ^@b\i seYSzf q^LÉVC%g
I ^~\É...q\~e> ^~\És-\eYCs E cCCEE |CCEE

```

O[ ; Q } yr は最初にウォームアップで調整する。ウォームアップのチェーンの数と本番のチェーンの数がそれぞれ ^~\ É...q\ ~e> ^~\ És-\ eYCs E cCCEE |CCEE 指定されている。さて [ ; [ ; を実行する。

```

c VCq^CY E ]}yrf\b@CYg
| \<\< E [; [; fVCq^CY> ^~\É...q\~eE^~\É...q\~e> ^~\És-\eYCsE^~\És-\eYCs g
{ \<\<i q^f q^LÉVC%6 †E†> %B%g
J \<\<i eqS^zÉs~\ - q%g

```

最後の eqS^zÉs~\ - q%g で以下のような事後分布サンプリングの要約が出力されたと思う。成功だ。

```

c mean std median 5.0% 95.0% n_eff r_hat
| phase 0.34 0.12 0.34 0.15 0.56 1363.51 1.00
{ sigma 0.34 0.06 0.33 0.24 0.42 1206.21 1.00
J Number of divergences: 0

```

以下では結果のチェックに -qfS を用いよう。[ ; [ ; は結果の健全性のチェックが重要である。-qfS は標準的なチェックが兼ね備えられているので便利である。

```

c S\ebqz -qfS<
| -qfS<i eYbzÉzq <Cf\<\<> f- qÉ^- \CsE9neP- sCn>nsSL\ - n: g

```

上では図Jiのように、パラメタの周辺分布（左）とチェーンが表示される。

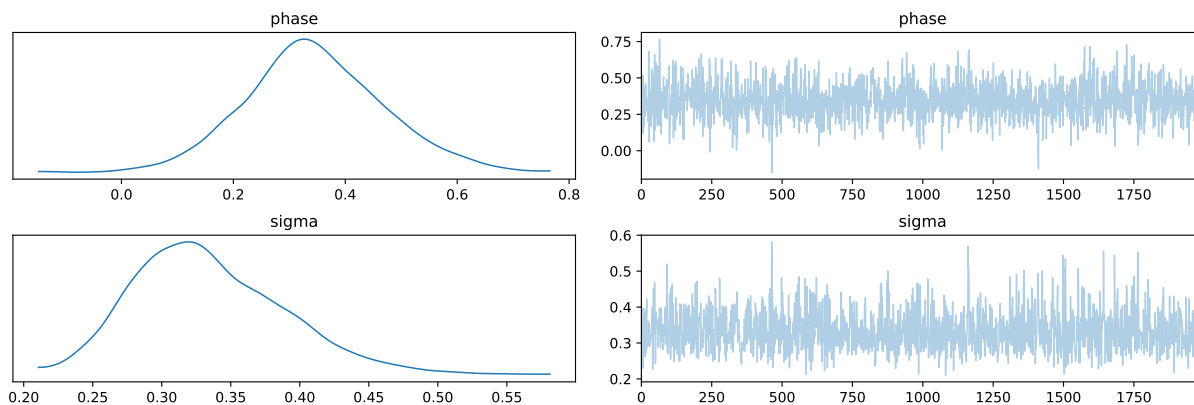


図4.2

いわゆるコーナープロットで事後分布の同時確率密度をチェックしよう<sup>1c</sup>。

<sup>1c</sup> | CFCc | 月現在 -qfS の @CfCbe 版を用いている。

```

c qCHbE78t qCHb9nsSL\ - nr: EsSLS^t qCHb9neP - sCn: EeP - sC
| - qfS<i eYbz€e - Sqf - qfS<i Hjb\€^~\e%qbf \<\>g>V8^@EoV@Co>
{ @SfCqLC^<CsEG YsC>\ - qLS^ - YsEyq - C>qCHCqC^<€€f - Y - CsEqCHb>
j qCHCqC^<€€f - Y - Cs€W.: qLsE7o<bYbqo=nqC@n> n\ - qVqpn=nbm> n\ - qVqpsS< Cn=c | 8g

```

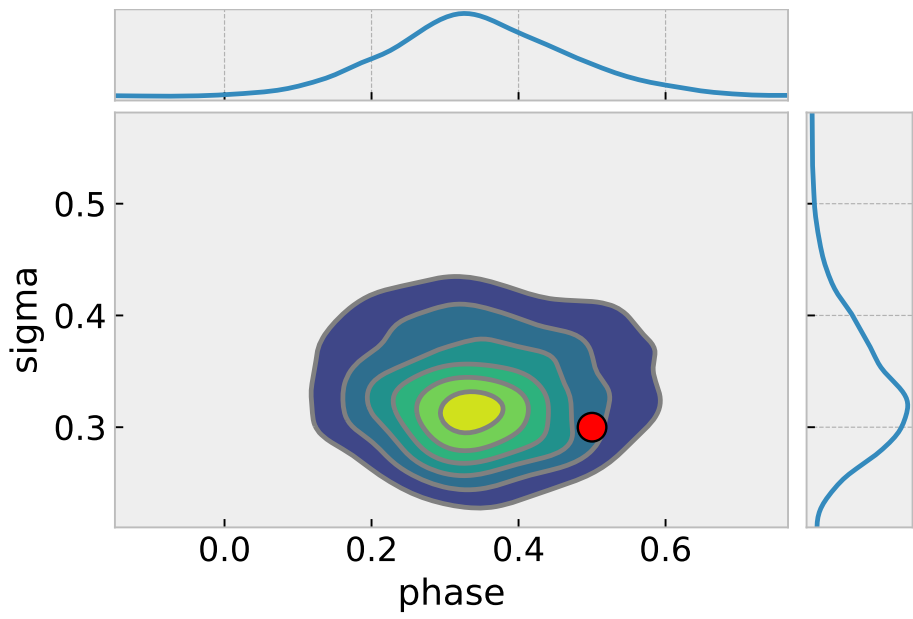


図4.3 赤い点は refs で指定されるインプットの値。

位相  $\phi$  および  $\sigma$  のサンプリングは

```

c ebszCqSbq€eP - sC E \<\<i LCz€s - \eYCsfg9oeP - sCo:
| ebszCqSbq€sSL\ - E \<\<i LCz€s - \eYCsfg9osSL\ - o:

```

のようにして得ることができる。この事後分布サンプリングを用いて、予測分布のサンプリングをつくろう。  
 $\sim \text{e}^{\text{qbi}^T \text{Hq}}$  の  $\text{dq}^{\text{C}^T \text{zSfC}}$  を用いることで簡単に予測分布をサンプルできる。

```

c Hjb\ ^~\e%qbi S^HCq S\ebqz dqC@S<zSfC
| eqC@ E dqC@S<zSfCf\b@CY>7oeP - sCo=ebszCqSbq€eP - sC>osSL\ - o=ebszCqSbq€sSL\ - 8>
{ qCz~q^€sSzCsE9n%a g
j ‡€ E U^ei YS^se - <Cf€J1U^ei eS>c€€€g
| eqC@S<zSb^s E eqC@f q^L€V%€> ‡E€€%€] b^Cg

```

信頼分布として  $\text{OSLPGsz ebszCqSbq}^{\text{C}^T \text{sS}^2 \text{s}^T \text{zCqf}} \text{Yod? Rg}$  を  $_{\text{C}}$  区間で計算しよう。

```

c Hjb\ ^~\e%qbi @S - L^bszS<s S\ebqz Pe@S
| \C - €€~%€E U^ei \C - ^feqC@S<zSb^s9n%a > - ‡Ss€€g

```

```
{ Pe@SE\~%oE Pe@SfeqC@S<zSb^s9n%a > CE_g
```

最後にデータとともにプロットしよう。もちろんプロットの見た目を良くするための `sG 4bq^` は使わなくても良い。結果は図4.4である。

```
c S\ebqz sG 4bq^ -s s^s
l eYzi sz%AC ~sCf o4\ Pog
{
j HSL> - † E eYzi s~4eYbzs f^qb.sEc> ^<bYsEc> HSLs< Cefv> {gg
i - †i eYbzf †>%aj n<bYbqEn4Y-<Wig
v - †i eYbzf †€>\C ^€\~%<bYbqEn; CEg
u - †i HSY€4Cz..CC^f †€> Pe@SE\~%CE> Pe@SE\~%a: > -YeP- ECE{> S^zCqebY- zCEyq~C<bYbqEn; CEg
d eYzi †Y- 4CYf n†ng
_ eYzi %f- 4CYf n%ag
```

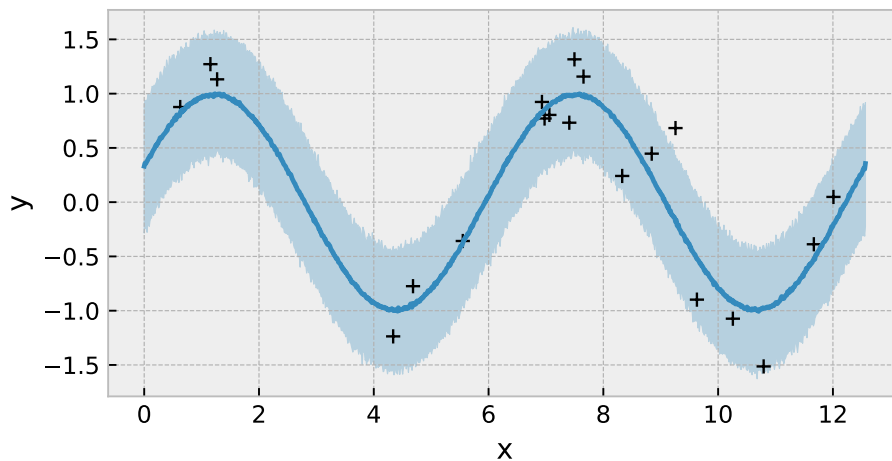


図4.4。

## dfbq

`P-^@Cp` を用いるとモデルからサンプリングできる。ただし、インプットに `%` は必要ないことに注意。

```
c @CH \b@CY<f†g=
l eP-sC E ^~\e%qi s- \eYCF oeP- sCo> @Szi } ^SHbq\ fQi CEU^ei eS> ci CEU^ei eSgg
{ sSL\^- E ^~\e%qi s- \eYCF osSL\^- o> @Szi B†eb^C^zS- Yf ci gg
j \~EU^ei sS^f †j eP-sCg
i @E^~\e%qi s- \eYCF o%o> @Szi | bq\^- Yf \~> sSL\^- gg
```



```
v | qCz~q^ @
```

これを  $P \sim \text{CpisCC}$  を用いてサンプリングできる。

```
c | Hjb\ ^~\e%qb S\ebqz P- ^@YCqs
| @ E P- ^@YCqsi sCC@f\b@CY<> q^LfsCC@E_gf†g
```

上の例では  $\backslash \text{b@CYf†g}$  の中で  $\wedge \sim \text{e} \text{ @pis } \backslash \text{e} \text{ @C}$  を用いて、データの確率分布を直接指定した。しかし場合によっては対数尤度からサンプリングしたいこともある。この場合、

```
c | @CH \b@CYf†>%g=
| eP- sC E ^~\e%qbi s- \eYCFoeP- sCo> @Sszi } ^SHbq\ fQi C EU^ei eS> ci C EU^ei eSgg
{ sSL\ - E ^~\e%qbi s- \eYCFosSL\ - o> @Sszi B†eb^C^zS- Yf ci gg
J \~EU^ei sS^f†j eP- sCg
| YbLYSWCYPbb@EQEI 1f%Q\~g11| vsSL\ - 11| Q^ei YbLf sSL\ - g
v ^~\e%qbi H <zbcf nYbLYSW†n> YbLYSWCYPbb@g
```

のようにして指定することもできる。

## Jicic

共通のパラメータを変数にもつ2つの種類のデータがある場合を考えよう。例として

$$\begin{aligned}
 y_1 &= sS^{\phi}(x + \phi) + \epsilon && \text{fJivg} \\
 y_2 &= \langle bs(x + \phi) + \epsilon && \text{fJiug} \\
 \epsilon &\sim \mathcal{N}(0, \sigma) && \text{fJiDg}
 \end{aligned}$$

という場合を考える。例えば図 JII のようなデータを解析する。

モデルは前節と同様に

$$\begin{aligned}
 y_1 &\sim \mathcal{N}(sS^{\phi}(x + \phi), \sigma) && \text{fJi\_g} \\
 y_2 &\sim \mathcal{N}(\langle bs(x + \phi), \sigma) && \text{fJicG}
 \end{aligned}$$

と考えることができるので、 $\wedge \sim \text{e} \text{ @p}$  のモデルは以下のように書ける。

```
c | @CH \b@CYf†c>†j>%>%g=
| eP- sC E ^~\e%qbi s- \eYCFoeP- sCo> @Sszi } ^SHbq\ fQi C EU^ei eS> ci C EU^ei eSgg
{ sSL\ - E ^~\e%qbi s- \eYCFosSL\ - o> @Sszi B†eb^C^zS- Yf ci gg
J \~cEU^ei sS^f†cj eP- sCg
| \~|EU^ei <bsf†j| eP- sCg
v ^~\e%qbi s- \eYCFo%>> @Sszi | bq\ - Yf\~c> sSL\ - g> b4sE%g
u ^~\e%qbi s- \eYCFo%>> @Sszi | bq\ - Yf\~|> sSL\ - g> b4sE%g
```

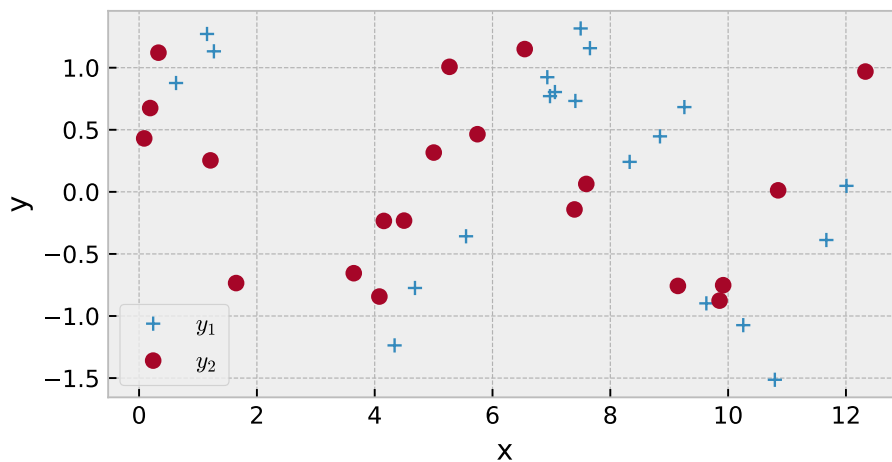


図4.5。

あとはほとんど前節と同じだが、[ ; [ ; 実行時に

```
\<\<i q^f q^L\%> \cE\c> \E\> \E\> \E\>
```

のように変数の数を合わせるのと、予測分布を描くときにも以下のように2変数にする注意が必要である。

```
eqC@ E dqC@S<zSfCf\b@CY>7oeP- sCo=ebszCqSbqfeP- sC>osSL\ - o=ebszCqSbqfeSL\ - 8
>qCz~q^EsSzCsE9n%an>n%an: g
\cE E U^ei YS^se- <CfCEJ1U^ei eS>cCEG
\cE E U^ei YS^se- <CfCEJ1U^ei eS>cCEG
eqC@S<zSb^s E eqC@f q^L\%>\cE\c>\E\>\E\>\E\> b^C>%E] b^Cg
```

予測からもそれぞれ平均・Od? Rを計算することにより、

```
\C ^E\~% E U^ei \C ^feqC@S<zSb^s9n%an> - \SsECg
Pe@SE\~% E Pe@SfeqC@S<zSb^s9n%an> > CE_g
\C ^E\~% E U^ei \C ^feqC@S<zSb^s9n%an> - \SsECg
Pe@SE\~% E Pe@SfeqC@S<zSb^s9n%an> > CE_g
```

最終的に、以下のようなプロットで

```
HSL> - \ E eYzi s~4eYbzf^qb. sEc> ^<bYsEc> HSLs< Cefv> {gg
- \i eYbzf \c>%>nj n<bYbqEn; Cfg
- \i eYbzf \c>\C ^E\~%<bYbqEn; Cfg
- \i HSYE4Cz. CC^f \c> Pe@SE\~%> Pe@SE\~%> - YeP- ECE{>
S^zCqebY- zCEyq-C<bYbqEn; Cfg
- \i eYbzf \>%>nbn<bYbqEn; cng
```

```

u - |i eYzbf | | €\C ^€\~%q><bYbqEn; cng
D - |i HSY€4Cz..CC^f | | €> Pe@S€\~%q9CE> Pe@S€\~%q9c: > - YeP- ECE{ >
- S^zCqebY- zCEyq~C><bYbqEn; cng
e€ eYzi |Y- 4CYf n|ng
cc eYzi %& 4CYf n%ag

```

図Jivのように同時フィットすることができる。

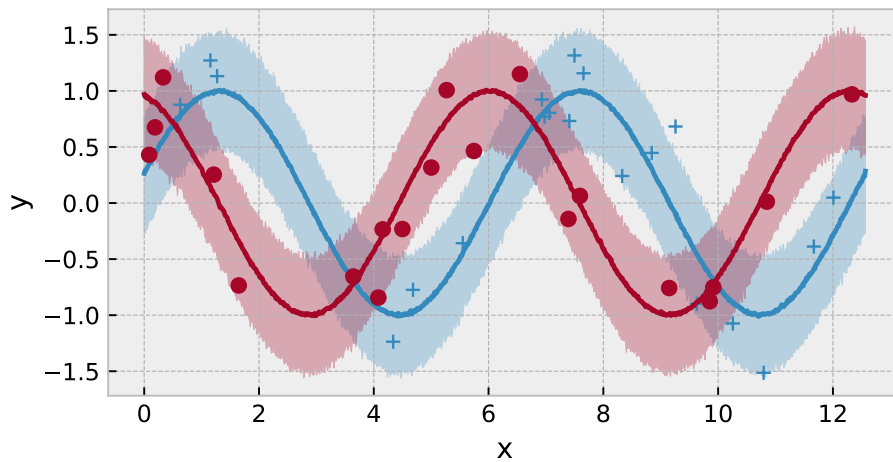


図4.6。

### Jici|

世の中はモデルは往々にして美しい数学関数だけで記述されるわけではない。複雑なモデルになるほど、実験データからもとまる関係などが入ってくる。このような場合でも補間関数を用いることによって対処できる。一例として、図Jiuの上パネルのようなデータ点（黒点）からモデルを構成することを考えよう。

このデータ点  $f \rightarrow \%o$  を  $U \ddot{i}^{\wedge} \sim e \% S^z C q e$  で補完した関数を以下のように定義する。

```

c @CH H f | g =
| qCz~q^ U^ei S^zCqef | > f - > %g

```

すると、 $f(x)$  は図Jiuの上パネル実線のように、データのある部分は良く補間されているのがわかる。さてこの  $H \ddot{i} g$  に対しても自動微分は通るので

```

c Hjb\ U | S\vebqz Lq @ f^ - e
| LEf^ - ef Lq @ f Hgg

```

のように定義すれば、微分を計算できる。図Jiuのしたパネルはそれを表示したもので、具体的には

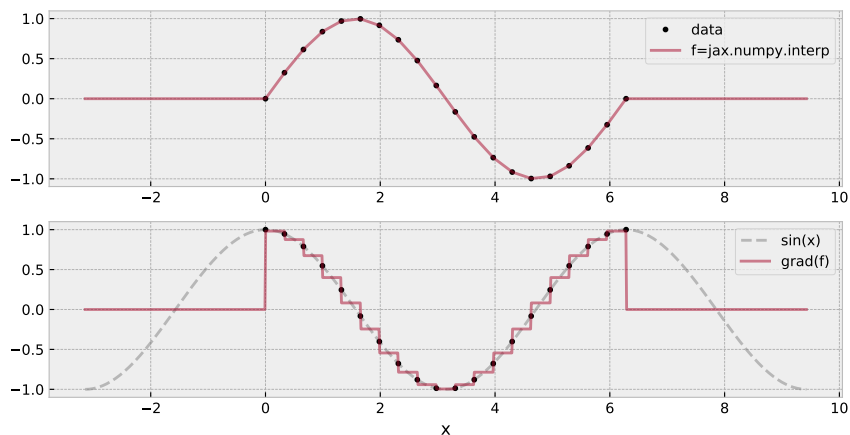


図4.7。

```

c 1^ei eS>cCEG
l eYzi eYzf 1^ei <bsf 1^g>YsEn@ sPC@>Y- 4CYEnsS^f 1^gn><bYbqEnLq- %a>- YeP- ECEI g
{ eYzi eYzf 1^ei <bsf 1^g>ni n><bYbqEn4Y- <Wig
J eYzi eYzf 1^ei <bsf 1^g>Y- 4CYEnLq- @fHgn><bYbqEn; cn>- YeP- ECEI g
l eYzi YCLC^@f g
v eYzi 1^Y- 4CYf n1^ng

```

で、作成されている。そう、実は（というほどでもないが）このデータ点は  $s^S(x)$  から作成されているのだが、そのことは知らないとして話を進めよう。補間関数の自動微分は、データ点の離散化に対応してカクカクしているものの、おおむね真のモデル  $\partial_x s^S x = \langle bs x$  を再現している。

さて補間関数  $f(x)$  を用いて以下のようにモデルを構成してみよう。

$$y = Af(x - c) + \epsilon \quad \text{fJiccg}$$

$$\epsilon \sim \mathcal{N}(0, \sigma) \quad \text{fJic|g}$$

上のモデルに  $c = 0.3, \sigma = 0.5, A = 10.0$  を用いて作成した模擬データを用いる。

```

c 1^4E^ei YS^se- <Cf CEI>ci u>cCG
l <ECE{
{ sSLS^ECEI
J , EcCECE
l %a 1^ei sS^f 1^4j <gj ^ei q- ^@bi ^bq\ - Yf CEsSLS^>sS< CEYC^f 1^4gg

```

したがって  $O[ ; Q ] yr$  のためのモデルは以下ようになる。

```

c S\ebqz ^~\e%qb

```

```

| S\ebqz ^~\e%qi @SszqS4~zSb^s -s @Ssz
{
J @CH\b@CYf†>%g=
i < E ^~\e%qis-\eYcf o<o> @Szi } ^SHbq\fqEI > CEI gg
v , E ^~\e%qis-\eYcf o, o> @Szi } ^SHbq\fc>cCEgg
u sSL\ - E ^~\e%qis-\eYcf osSL\ - o> @Szi B\eb^C^zS- Yf ci Cgg
D \~E, 1Hf†j <g
- ^~\e%qis-\eYcf o%> @Szi ] bq\ - Yf\~> sSL\ - g> b4sE%g

```

これで、 $O[; Q]_{yr}$  をいつものように回すと、ちゃんと推定ができる。今回は図JiDに  $q@S4YCS^zCqf-Y$  だけ示そう。

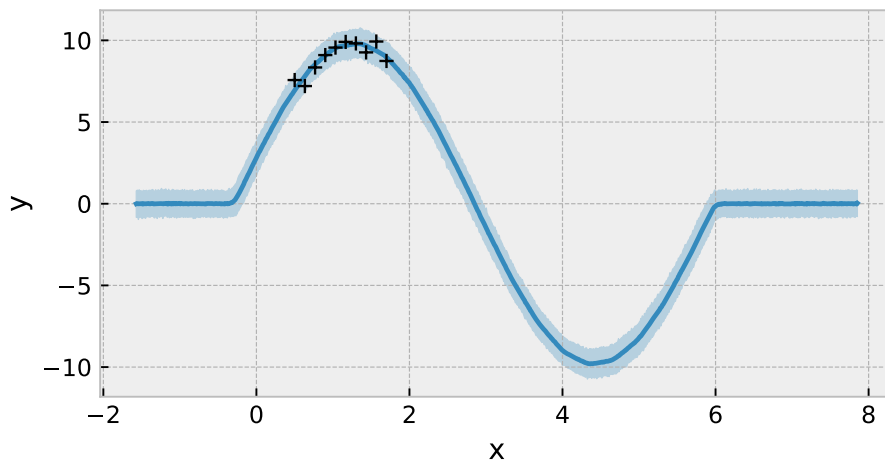


図4.8。

### Jici{

次の例は、畳み込みの幅を推定する問題である。これは例えばあるシャープなシグナルが、なんらかの過程でなまされてしまったものが観測されるときなどに有用である。典型的なのは観測装置由来のレスポンスである。このような場合  $U\ddagger\wedge\sim e\%b^f\%Y\%C$  を用いることができる。

レスポンスのカーネルとして次のようなガウス関数型を考えよう。なまされる幅は  $\sigma = sLW$  でパラメトライズされるとする。図Jiに、なまされかたの違いを示す。

さてデータとしてはデルタ関数的なピークが何本かたっているもの  $f\%g$  を用意して、これをカーネルでなましたものを用いる。 $O[; \text{推定}$  としては以下のように構成できる。

```

c | WcCEE
| †WEU^ei YS^se- <CfQEC>CEc>] Vg
{

```

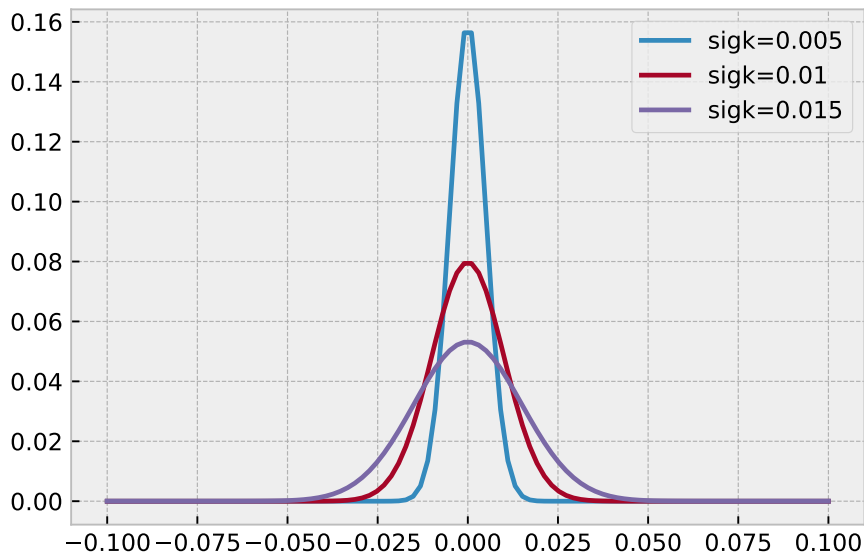


図4.9。

```

j @CH \b@CYf †>%g=
i sSLWE ^~\e%pbis-\eYCF osSLWb> @Sszi B†eb^C^zS- Yf ci gg
v sSL\ - E ^~\e%pbis-\eYCF osSL\ - o> @Sszi B†eb^C^zS- Yf ci gg
u \~EU^ei <b^fbYfCf%>U^ei C†ef Q†W11|wf|i lsSLW1|gg 5
d wU^ei sl qzf|i 1U^ei eSgvsSLWf C†|w|Vg>ns-\Cng
- ^~\e%pbis-\eYCF o%> @Sszi |bq\ - Yf\~> sSL\ - g> b4sE%g

```

結果は図JicDの通り。

Ji|  $\hat{\epsilon}_i$

pCfCpC\ b@C

$y_i$  はデフォルトでは  $pCfCpC\ b@C$  を使用する。そこでカスタマイズした自動微分を用いるためには第 |iJi| 章で説明した、Td を定義したものを用いればよい。ここでは  $f(x, A) = A S x$  の元に、 $y_i = f(x_i, A) + \epsilon_i$ 、および  $\epsilon_i \sim \mathcal{N}(0, \sigma)$  というモデルで  $y = \{y_i\}$  をフィットしよう。Td の定義は

```

c 2<-szb\€fUe
l @CH Pf †>, g=
{ qCz~q^ , 1U^ei sS^f †g
j
i @CH PEH.@ †>, g=

```

{u

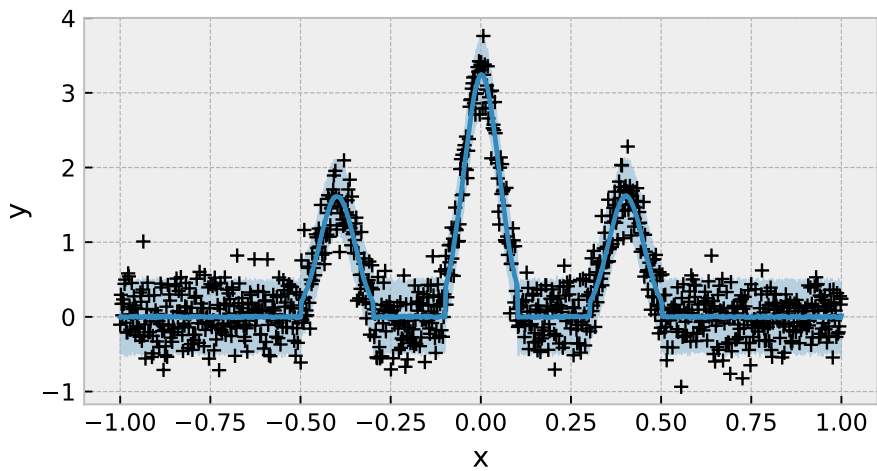


図4.10。

```

v      qCs E f, 1U^ei <bsf †g> U^ei sS^f †gg
u      qCz~q^ Pf †, g> qCs
D
-      @CH PE4. @qCs> ~g=
cOE    , €<bs€†> sS^€† E qCs
cc     qCz~q^ f, €<bs€† 1 ~> sS^€† 1 ~g
cl
cl     Pi @Ch/UefPEH. @ PE4. @g

```

となる。モデルは

```

c      @CH \b@CYf †>%g=
|      sSL\ - E ^~\e%qbi s- \eYCF osSL\ - o> @Szi B†eb^C^zS- Yf ci gg
{      †CEE ^~\e%qbi s- \eYCF o†CE> @Szi } ^SHbq\ f Qi > ci gg
J      , E ^~\e%qbi s- \eYCF o, o> @Szi B†eb^C^zS- Yf ci gg
I      PfEf\ - efP>f CE| b^Cg> CG
v      \~EPff ††CE, g
u      ^~\e%qbi s- \eYCF o% @Szi | bq\ - Yf \~> sSL\ - g> b4sE%g

```

とすればよい。ここに  $f$  -e を用いて  $\mathbf{x}$  に対してベクトル化していることに注意。  $O[ ; Q ]$  yr は

```

c      q^L€VC%oE q ^@b\i dp| KVC%foCG
|      q^L€VC%o q^L€VC%6 E q ^@b\i seYSzf q^L€VC%g
{      ^~\€.. q\~e> ^~\€s- \eYCs E cCEE |CEE
J      Vq^CY E }}yrf\ b@CYg

```

i	\<\< E [ ; [ ; fVq^CY> ^~\€.:q\~e> ^~\€s-\eYCsg
v	\<\<i q~^fq^L€V%€> †E†> %€@ z-g
u	\<\<i eqS^z€s~\\- q%€g

とすればよい。

**Gbq.:q@\ b@C**

] ~\ d%€p €i€€より Hbq.:q@\ b@Cでの O[ ; ] } yr も動作するようになった。



# I

## ] ~ \ e % \phi

### I ic

ガウス過程は、多変数正規分布

$$\mathbf{d} \sim \mathcal{N}(\mathbf{0}, \Sigma) \quad \text{flicg}$$

に従う確率変数  $\mathbf{x}$  の確率過程のことである。いま  $d_i$  は時系列だとして、この多変数正規分布の共分散行列の各成分が、

$$\Sigma_{ij} = K_{ij}(a, \tau) = ak(|t_i - t_j|; \tau) \quad \text{fliig}$$

のように、 $t_i$  と  $t_j$  の差の絶対値の関数として与えられれば、相関長  $\tau$  のガウス過程が得られる。カーネル関数  $k(t, \tau)$  としてはいろいろなタイプがありうるが、例えば p3G カーネル

$$k_{p3G}(t; \tau) = Cte^{-\frac{t^2}{2\tau^2}}, \quad \text{fliig}$$

と [-z q^ {w} カーネル

$$k_{\{-z\}}(t; \tau) = 1 + \frac{\sqrt{3}t}{\tau} e^{-\sqrt{3}t/\tau}. \quad \text{fliJg}$$

などがある。

とりあえず論より実装で、例を作ってみよう。上記カーネルは

```

c S\ebqz ^~\e% s ^e
| @CH p3Gfz>z- ~g=
{   ?z E z Q ^ei - qq- %fz: gi y
J   VE^ei Ctef Cf?zg11|w|wfz- ~11| gg
I   qCz~q^ V
v
u @CH [-zCq^{|fz>z- ~g=
D   ?z E z Q ^ei - qq- %fz: gi y
-   H <E^ei sl qzf{ i Cg1^ei - 4sf?zgwz- ~

```

```

cE Vef ci QjH <g1^ei Cjef QH <g
cc qCz~q^ V

```

のような感じである。さて、 $s \sim \mathcal{N}(z, \Sigma)$ 、 $y \sim \mathcal{N}(f(z), \sigma^2)$  -Yで多変数正規分布がサンプリングできる<sup>1c</sup>。

```

c Hjb\ s<Se%sz-zs S\ebqz \~YzSf- qS- zC^bq\ - Y -s s\^
| ^ei q- ^@b\i sCC@f sCC@Ecg `
{ ] E cE
j z E ^ei YS^se- <Cf CcE] g
r - fC E ^ei < Cqbsf ] g
v z- ~ E CEJ
u - Eci CE
D - fC E U^ei < Cqbsf ] g
- <bf E - 1p3Gfz>z- ~g ` <bf E [-zCq\} /fz>z- ~g
cE @S E s\^f\G ^E- fC ><bfE<bf > - Yyb. fS^L~Y- q Eyq-Cgi qfsfcgi y

```

さて、このように生成されたデータに平均ゼロ、標準偏差  $\sigma$  のガウスノイズを足しておく。このノイズを便宜的に観測ノイズと呼んでおこう。

```

c sSL\ - ECEv
| @E@Sj ^ei q- ^@b\i ^bq\ - Yf CECEsSL\ - >YC^f @Sgg

```

図Iicが生成されたデータである。

さてこのデータを元に  $\tau$ 、 $a$ 、 $\sigma$  を推定してみよう。平均ゼロ、標準偏差  $\sigma$  のガウスノイズの観測ノイズを足したということは確率モデルとしては、

$$\begin{aligned}
 \mathbf{d} &\sim \mathcal{N}(\mathbf{0}, \Sigma') && \text{fiilg} \\
 \Sigma' &= K(a, \tau) + \sigma^2 I && \text{fiivg}
 \end{aligned}$$

となる。この  $\Sigma'$  を以下のように実装する。

```

c @CH \b@CY<bfz>z- ~>- >sSL\ - g=
| ?z E z Q U^ei - qq- %fz: gi y
{ VE- 1U^ei Cjef Qf?zg11| w| wfz- ~11| ggj U^ei C%Gf] g1sSL\ - 11|
j qCz~q^ V

```

$\mathcal{N}(z, \Sigma)$  のモデルはこの  $\mathcal{N}(z, \Sigma)$  を用いて

```

c S\ebqz U j i ^~\e%o s U^e
| S\ebqz ^~\e%b
{ S\ebqz ^~\e%bi @SszqS4~zSb^s -s @Ssz

```

<sup>1c</sup>  $\mathcal{N}(z, \Sigma)$  や  $\mathcal{N}(z, \Sigma)$ 、および  $\mathcal{N}(z, \Sigma)$ 、 $\mathcal{N}(z, \Sigma)$  -Yでもあるのだがとある理由 fvic章参照によりここでは  $\mathcal{N}(z, \Sigma)$  のものを用いる。

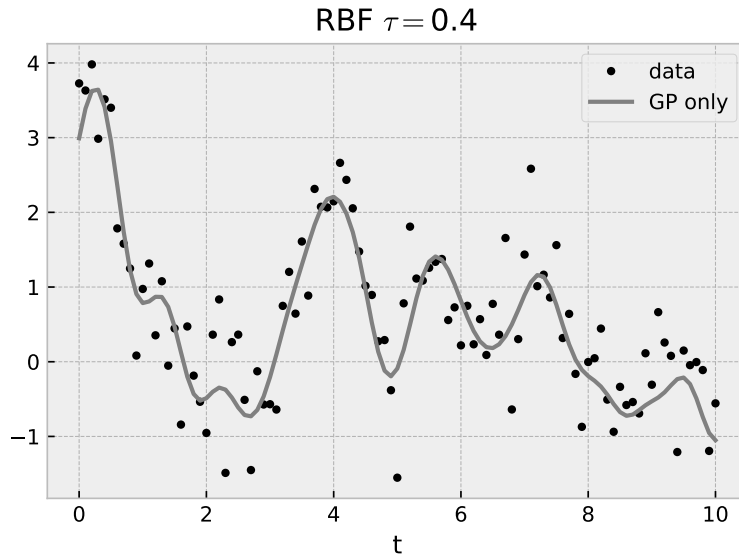


図5.1 実線は  $\sigma$  の観測ノイズなし。

```
J
I @CH \b@CYfz>%g=
v sSL\ - E ^~\e%qbi s- \eYCFosSL\ - o> @Sszi Bteb^C^zS- Yf ci gg
u z- ~ E ^~\e%qbi s- \eYCFoz- ~o> @Sszi Bteb^C^zS- Yf ci gg
D - E ^~\e%qbi s- \eYCFo-o> @Sszi Bteb^C^zS- Yf ci gg
_ <bf E \b@CY<bffz>z- ~-> >sSL\ - g
c@E ^~\e%qbi s- \eYCFo%o> @Sszi [ ~YzSf- qS- zC] bq\ - Yf Yb<EU^ei <Cqbsf ] g>
cc <bf- qS- ^<C€\ - zqSfE<bf> b4sE%g
```

となる。ここでは  $\tau$  と  $\sigma$  の事前分布を  $Bteb^C^zS$  分布とした。  $^~\e%q$  の  $O[ rQ ]$  yr を以下のように回す。

```
c Hjb\ ^~\e%qbi S^HCq S\ebqz [; [; > ]}yr
l Hjb\ U ‡ S\ebqz q- ^@b\
{ q^L€V%oE q- ^@b\i dp] KVC%oG
J q^L€V%o q^L€V%o E q- ^@b\i seYSzf q^L€V%o
I ^~\€..q\~e> ^~\€s- \eYCs E c€€€€ |€€€€
v ` p-^ ]}yri
u Vq^CY E ]}yrf\b@CYg
D \<\< E [; [; fVQ^CY> ^~\€..q\~e> ^~\€s- \eYCs g
_ \<\<i q-^fq^L€V%o zEz> %o@g
```

そして

```

c S\ebqz - qfS<
| qCH5E78t qCH59nsSL\ - n: EsSL\ - t qCH59nz - ~n: Ez - ~t qCH59n - n: E-
{ - qfS< i eYbz€€ - Sqf - qfS< i Hjb\€^\e%qbf\<\<g>V8^@EoW@Co>
j @SfCqLC^<CsEG YsC>\ - qLS^ - YsEyq - C> qCHCqC^<C€f - Y - CsEqCH5>
i qCHCqC^<C€f - Y - Cs€W.: qLsE7o<bYbqo=nqC@n> n\ - qWqn=nbm> n\ - qWqsS: Cn=c | 8g

```

のように事後分布を可視化したものが図5.2である。ところで、式(5.1)という観点のモデル化では、 $\mathbf{x} = \mathbf{0}$

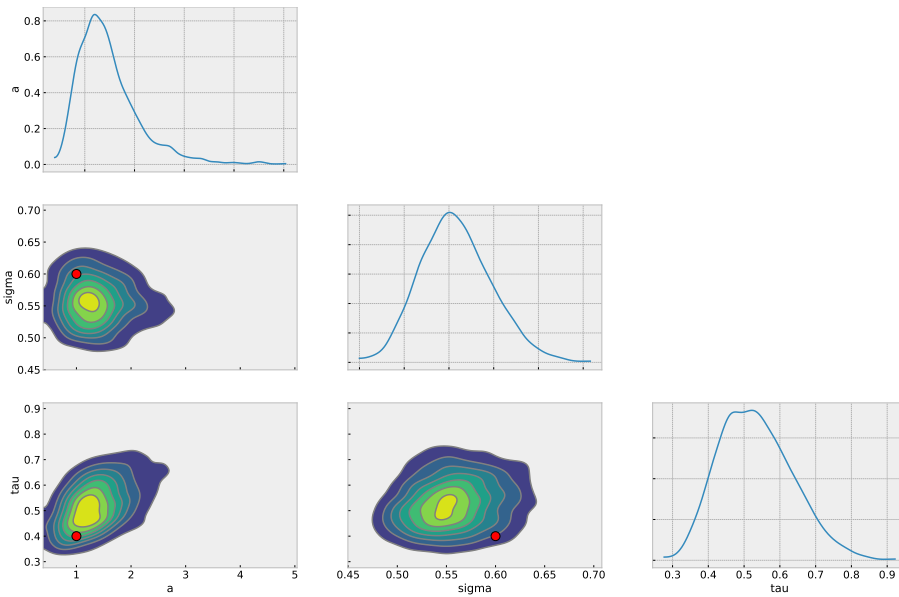


図5.2

がモデル平均である。これは第4章で行ったのと同様に  $\int \mathbf{q}(\mathbf{z}|\mathbf{y})$  を計算することでより明確になる。つまり

```

c Hjb\ ^~\e%qbi S^HCq S\ebqz dqC@S<zSfC
| Hjb\ ^~\e%qbi @S - L^bszS<s S\ebqz Pe@S
{
j ebszCqSbq€€ E \<\<i LCz€s - \eYCsf g9o - o:
i ebszCqSbq€z - ~ E \<\<i LCz€s - \eYCsf g9oz - ~o:
v ebszCqSbq€sSL\ - E \<\<i LCz€s - \eYCsf g9osSL\ - o:
u
D eqC@ E dqC@S<zSfC\ b@CY>7o - o=ebszCqSbq€€ >oz - ~o=ebszCqSbq€z - ~>
- osSL\ - o=ebszCqSbq€sSL\ - 8>qCz - q^€sSzCsE9n%aa g
c€€ eqC@S<zSb^s E eqC@f q^L€W%6>zEz>%aa b^Cg
cc \C - ^€~%aaE U^ei \C - ^feqC@S<zSb^s9n%aa > - †SsECCg
c| Pe@S€~%aaE Pe@SfeqC@S<zSb^s9n%aa > C€_g

```

で予測サンプリングから  $P_{t|S}$  を計算し、

```

c HSL> - ‡ E eYzi s~4eYbzf ^qb.sEc> ^<bYsEc> HSLs< Cefv> {gg
l - ‡i eYbzfz>@ni n><bYbqEn4Y<Wlg
{ - ‡i eYbzfz€>\C ^€\~%<bYbqEn; Cfg
J - ‡i HSY€4Cz..CC^fz€> Pe@S€\~%@CE> Pe@S€\~%@c: > -YeP- ECE{> S^zCqebY- zCEyq~C><bYbqEn; Cfg
l eYzi ‡Y- 4CYf nznng
v eYzi %f- 4CYf n%ag
  
```

のようにプロットすると、図5.3となり、直感的にわかる。

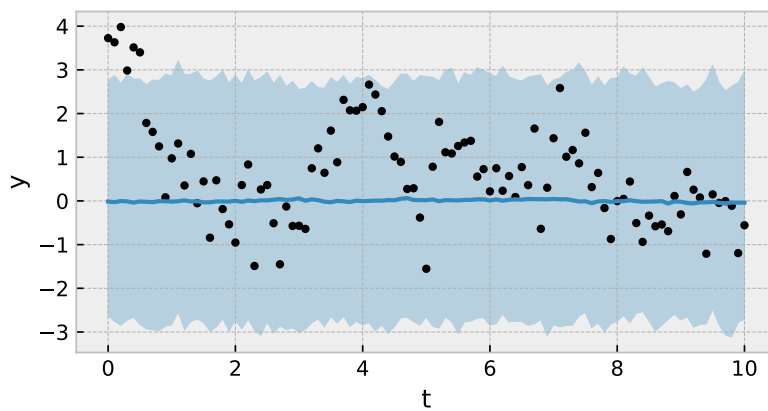


図5.3

上記のガウス過程は、モデルパラメタ  $\mathbf{m}$  の事前分布（プライア）として

$$p(\mathbf{m}) = \mathcal{N}(\mathbf{0}, \Sigma) \quad \text{fliug}$$

と置き、

$$\begin{aligned} d_i &= m_i + \epsilon && \text{fliDg} \\ \epsilon &\sim \mathcal{N}(0, \sigma^2) && \text{fli_g} \end{aligned}$$

と見ることできる。ここに  $\epsilon$  は観測ノイズに対応している。もう少しまどろっこしく書くとモデル  $\mathbf{g}$  を恒等変換として、

$$\begin{aligned} \mathbf{g}(\mathbf{m}) &= \mathbf{m} && \text{flicG} \\ \mathbf{d} &= \mathbf{g}(\mathbf{m}) + && \text{fliccg} \\ &\sim \mathcal{N}(0, \sigma^2 I) && \text{flic|g} \end{aligned}$$

のように書ける。この場合、尤度関数が

$$p(\mathbf{d}|\mathbf{m}) = \mathcal{N}(\mathbf{d} - \mathbf{g}(\mathbf{m}), \sigma^2 I) = \mathcal{N}(\mathbf{d} - \mathbf{m}, \sigma^2 I) \quad \text{flic\{g}$$

となる。そして、さらにこの事前分布の中にパラメタがあるという構造となっている。つまり

$$\Sigma_{ij} = ak(|t_i - t_j|; \tau) \quad \text{flicJg}$$

としていて、このパラメタ  $a, \tau$  をハイパーパラメタという。そしてハイパーパラメタの事前分布  $f$  超事前分布またはハイパープライア  $g$  をさらに仮定するということに相当する。

以上のモデルを、直接  $\mathcal{N}(\mathbf{m} | \mathbf{d}, \Sigma)$  のモデルとして構成し、 $\mathbf{m}$  の事後分布をサンプリングすることもできるが、ハイパーパラメタを固定した状態では、 $\mathbf{m}$  の事後分布を解析的に書けることが知られている。ここでガウス過程の計算法を紹介しよう。まず多変数正規分布に多変数正規分布をかけても多変数正規分布である。そこで計算しなくてはならないのは  $\mathcal{N}(\mathbf{d} | \mathbf{m}, \sigma^2 I)$  のべきの部分のみである。多変数正規分布の「べき」部分は

$$-2 \mathcal{Y}L \mathcal{N}(\mathbf{m} | \mathbf{d}, \Sigma) = (\mathbf{m} - \mathbf{d})^\top \Sigma^{-1} (\mathbf{m} - \mathbf{d}) = \mathbf{m}^\top \Sigma^{-1} \mathbf{m} - 2 \mathbf{m}^\top \Sigma^{-1} \mathbf{d} + \mathbf{d}^\top \Sigma^{-1} \mathbf{d} \quad \text{fliclg}$$

となっていることに注意すると、もしある多変数正規分布に従うことがわかっている確率密度分布  $p(\mathbf{m})$  が

$$-2 \mathcal{Y}L p(\mathbf{m}) = \mathbf{m}^\top P \mathbf{m} - 2 \mathbf{m}^\top \mathbf{q} + \mathbf{q}^\top P \mathbf{q} \quad \text{flicvg}$$

のように書けたとすると、式  $\text{fliclg}$  と比べることで、この確率密度分布は

$$p(\mathbf{m}) = \mathcal{N}(\mathbf{m} | P^{-1} \mathbf{q}, P^{-1}). \quad \text{flicug}$$

であることがわかる。

さて、いま事後分布は

$$p(\mathbf{m} | \mathbf{d}) \propto p(\mathbf{d} | \mathbf{m}) p(\mathbf{m}) = \mathcal{N}(\mathbf{d} - \mathbf{m}, \sigma^2 I) \mathcal{N}(\mathbf{0}, \Sigma) \quad \text{flicDg}$$

であるので、この「べき」部分を  $\mathbf{m}$  について展開することで、

$$-2 \mathcal{Y}L [\mathcal{N}(\mathbf{d} - \mathbf{m}, \sigma^2 I) \mathcal{N}(\mathbf{0}, \Sigma)] = \mathbf{m}^\top (\Sigma^{-1} + \sigma^{-2} I) \mathbf{m} - 2 \sigma^{-2} \mathbf{m}^\top \mathbf{d} + \mathbf{d}^\top \Sigma^{-1} \mathbf{d} \quad \text{flic_g}$$

と書けること、また

$$(\Sigma^{-1} + \sigma^{-2} I)^{-1} = \Sigma (I + \sigma^{-2} \Sigma)^{-1} \quad \text{flicG}$$

より

$$p(\mathbf{m} | \mathbf{d}) = \mathcal{N}(\Sigma (\sigma^2 I + \Sigma)^{-1} \mathbf{d}, \Sigma (I + \sigma^{-2} \Sigma)^{-1}) \quad \text{fliccg}$$

であることがわかった。

さて、先程、行った  $\mathcal{O}(\mathbf{d}; \mathbf{Q})$  による  $\tau$  と  $\sigma$  のサンプリングがこの枠組みで何を意味するか考えよう。これらは今の枠組みではハイパーパラメタとみなせるので、まとめて  $\mathbf{a} = (a, \tau, \sigma)^\top$  をハイパーパラメタとおく。ここからは  $\mathbf{a}$  も確率に入れて考えることにする。さて、今、 $\mathbf{m}$  について周辺化した尤度

$$p(\mathbf{d} | \mathbf{a}) = \frac{p(\mathbf{d} | \mathbf{m}, \mathbf{a}) p(\mathbf{m}, \mathbf{a})}{p(\mathbf{m} | \mathbf{d}, \mathbf{a})} \quad \text{flicg}$$

はやはり多変数正規分布となるが、また「べき」部分の  $\mathbf{d}$  に関する部分だけ考えることにより

$$\begin{aligned} -2 \mathcal{Y}L p(\mathbf{d} | \mathbf{a}) &= -2 \mathcal{Y}L p(\mathbf{d} | \mathbf{m}, \mathbf{a}) + 2 \mathcal{Y}L p(\mathbf{m} | \mathbf{d}, \mathbf{a}) + \mathbf{d}^\top \Sigma^{-1} \mathbf{d} & \text{flic\{g} \\ &= -2 \mathcal{Y}L \mathcal{N}(\mathbf{d} - \mathbf{m}; \sigma^2 I) + 2 \mathcal{Y}L \mathcal{N}(\Sigma (\sigma^2 I + \Sigma)^{-1} \mathbf{d}, \Sigma (I + \sigma^{-2} \Sigma)^{-1}) & \text{flicJg} \\ &= \mathbf{d}^\top (\sigma^2 I + \Sigma)^{-1} \mathbf{d} + \mathbf{d}^\top \Sigma^{-1} \mathbf{d} & \text{flicIg} \end{aligned}$$

である。よって

$$p(\mathbf{d} | \boldsymbol{\tau}) = \mathcal{N}(\mathbf{0}, \Sigma + \sigma^2 I) \quad \text{fl i | v g}$$

となる。これは  $\Sigma = K(\boldsymbol{\tau})$  とすれば、式 fl i l g と同じである。すなわち、(超) 事前分布として  $p(\boldsymbol{\tau})$  を与えた時の周辺化事後分布

$$p(\boldsymbol{\tau} | \mathbf{d}) \propto p(\mathbf{d} | \boldsymbol{\tau}) p(\boldsymbol{\tau}) \quad \text{fl i | u g}$$

を  $O[; \mathbf{Q}]_{\text{yr}}$  でサンプリングしていたことになり、fl i i はハイパーパラメタ  $\boldsymbol{\tau}, \sigma$  の周辺化事後分布を示していることがわかった。つまり  $k = 0, \dots, N_s - 1$  に対して

$$\boldsymbol{\tau}_k^\dagger \sim p(\boldsymbol{\tau}_k | \mathbf{d}) \propto p(\mathbf{d} | \boldsymbol{\tau}_k) p(\boldsymbol{\tau}_k) \quad \text{fl i | D g}$$

をサンプリングしたことになる。さて、ではこのサンプルと式 fl i | c g を用いて、

$$p(\mathbf{m}, \boldsymbol{\tau} | \mathbf{d}) = p(\mathbf{m} | \boldsymbol{\tau}, \mathbf{d}) p(\boldsymbol{\tau} | \mathbf{d}) \quad \text{fl i | _ g}$$

であることから

$$\begin{aligned} p(\mathbf{m}_k^\dagger | \boldsymbol{\tau}_k^\dagger, \mathbf{d}) &= \mathcal{N}(\mathbf{m}_k^\dagger, K_k) & \text{fl i | C g} \\ K_k &= K(a_k^\dagger, \tau_k^\dagger) ((\sigma_k^\dagger)^2 I + K(a_k^\dagger, \tau_k^\dagger))^{-1} \mathbf{d} & \text{fl i | c g} \\ K_k &= K(a_k^\dagger, \tau_k^\dagger) (I + (\sigma_k^\dagger)^{-2} K(a_k^\dagger, \tau_k^\dagger))^{-1} & \text{fl i | g} \end{aligned}$$

からサンプルした  $\mathbf{m}_k^\dagger$  と  $\boldsymbol{\tau}_k^\dagger$  のセットは

$$(\mathbf{m}_k^\dagger, \boldsymbol{\tau}_k^\dagger) \sim p(\mathbf{m}, \boldsymbol{\tau} | \mathbf{d}) \quad \text{fl i | { g}$$

とみなせることがわかる<sup>1)</sup>。式 fl i | c g fl i | g を以下のように実装する。

```

c @CH \~Kdfz- ~> >sSL\ - g=
l <bf E - 1p3Gfz>z- ~g
{ RV.EsSL\ - 11| 1^ei C%Gf] gi <bf
J , Es<Se%oS^ - YLi sbYfCf RV.>@- ss-\C€ Enebsng
I qCz~q^ <bf2
v
u @CH <bfKdfz- ~> >sSL\ - g=
D <bf E - 1p3Gfz>z- ~g
_ RV.E^ei C%Gf] gi <bfvsSL\ - 11|
cE RV.Es<Se%oS^ - YLi S^ff RV.g
cc qCz~q^ <bf2RV...

```

のように定義する。次に式 fl i | C gからの  $\mathbf{m}$  のサンプリングを

1) ハイパーパラメタの点推定 (いわゆる  $\hat{\boldsymbol{\tau}}, \hat{\sigma}$  を用いる方法については解説が多くあるが、周辺化事後分布から再サンプリングする議論についてはいまのところ文献が見つからない。ベイズ線形問題+ガウス過程の枠組みで以前、筆者が議論したことがあるので参考までに挙げておく  $\mathbf{q}$ 。

```

c S\ebqz zl @
l ]sEYC^f ebszCqSbq€sSL\ - g
{ ^ei q- ^@b\i sCC@f sCC@Ecg
j \- qqE9:
r Hbq S S^ zl @i zl @f q- ^LCf €] sgg=
v sSL\ - sEHB- zf ebszCqSbq€sSL\ - 9S: g
u z- ~sEHB- zf ebszCqSbq€z- ~9S: g
D - €sEHB- zf ebszCqSbq€ 9S: g
- - fCE\ -Kdf z- ~s>- €s>sSL\ - sg
c€ <bfe<bfKdf z- ~s>- €s>sSL\ - sg
cc \WE s\^f\C ^E- fC ><bfe<bf > - Yyb.€sS^L~Y- q Eyq-Cgi qfsf cgi y
ei \- qqi - eeC^@f \Wg
cf \- qqE^ei - qq- %€\ - qqg

```

のように実行し、

```

c \C ^€\~%aE ^ei \C ^f\ - qq> - †SsECg
l Pe@S€\~%aE Pe@Sf\ - qq> CE_g

```

Od? Rを計算することで、図1.1の  $m$  の  $\langle q^T S^T S^T z C q^T Y \rangle$  が求まる。ここで注意点としては、これはモデル  $m$  の  $\langle E^T S^T z C q^T Y \rangle$  であり、観測ノイズ部分を含んだ  $d$  の予測ではないという点である。

```

c HSL EeYzi HSL~qCf HSLsS< Cef v> { gg
l - †EHSLi - @€s~4eYbzf cccg
{ - †i eYbzf z>@ni n><bYbqEn4Y- <Wig
j - †i eYbzf z> \C ^€\~%a<bYbqEn; cng
r - †i HSY€4Cz.CC^fz> Pe@S€\~%aCE> Pe@S€\~%aC: > - YeP- ECE{> S^zCqebY- zCEyq-C<<bYbqEn; cng
v eYzi †Y- 4CYf nznng
u eYzi %€ 4CYf n%ag

```

さて、一般に  $t = t^*$  での予測値はどうなるだろうか？以下では再度ハイパーパラメタを省略して表記する。まず、 $m$ 、 $m^*$  がガウス過程に従うとき

$$p(m^* | m) = \mathcal{N}(K_{\times}^{\top} K^{-1} m, K_* - K_{\times}^{\top} K^{-1} K_{\times}) \quad \text{fl i\{Jg}$$

である f, eeC^@f, g ここに

$$\begin{aligned}
K_{ij} &= ak(|t_i - t_j|; \tau) && \text{fl i\{Ig} \\
(K_{\times})_{ij} &= ak(|t_i - t_j^*|; \tau) && \text{fl i\{vg} \\
(K_*)_{ij} &= ak(|t_i^* - t_j^*|; \tau) && \text{fl i\{ug}
\end{aligned}$$



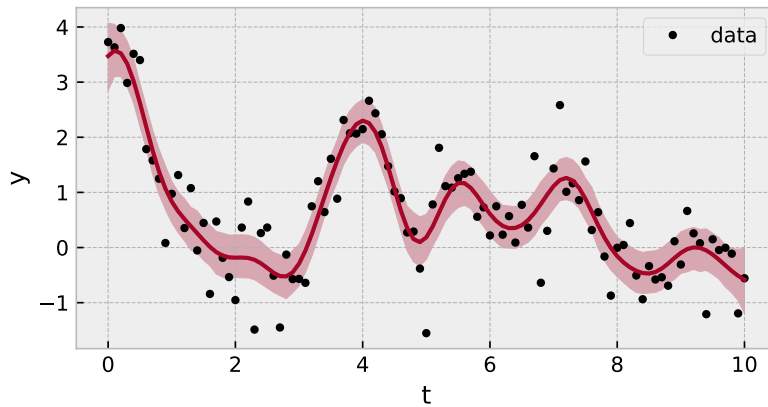


図5.4

同様に

$$p(\mathbf{m}^* | \mathbf{d}) = \mathcal{N}(K_{\times}^{\top} K^{-1} \mathbf{d}, K_{*} - K_{\times}^{\top} K^{-1} K_{\times}) \quad \text{fl i } \{Dg$$

となる。ここに

$$(K_{\times})_{ij} = ak(|t_i - t_j|; \tau) + \sigma^2 \delta_{ij} \quad \text{fl i } \{g$$

$\delta_{ij}$  はクロネッカーデルタ  $g$ 。ここで  $p(\mathbf{m}^* | \mathbf{d})$  はあくまでモデルパラメタとしてのガウス過程の事後分布なので、観測ノイズの分は考慮されていない。

もし観測ノイズを含んだ予測を行いたいならば、

$$\mathbf{d}^* = \mathbf{m}^* + \quad \text{fl i } J \{g$$

というモデルに基づき、

$$p(\mathbf{d}^* | \mathbf{d}) = \mathcal{N}(K_{\times}^{\top} K^{-1} \mathbf{d}, K_{*}; - K_{\times}^{\top} K^{-1} K_{\times}) \quad \text{fl i } Jcg$$

となるだろう。ここに

$$(K_{*};)_{ij} = ak(|t_i^* - t_j^*|; \tau) + \sigma^2 \delta_{ij} \quad \text{fl i } J |g$$

である。

というわけで、ハイパーパラメタ込みのサンプリングで式  $\text{fl i } \{D i Jcg$  のサンプリングを行い ;  $q \{SAC$   $RzCqf$ - $Y$  を求める。

```

c @CH \~<bfKd\#fz>z@z- ~> >sSL\ - g=
| <bf E - 1p3Gfz>z- ~g j sSL\ - 11| 1^ei C%cf] g
{ <bf\#E - 1p3G\#fz>z@z- ~g
J ` \
| <bf\#\# E - 1p3Gfz@z- ~g
v `

```

```

u | <bf†† E - 1p3Gfz@z- ~g j sSL\ - 1| 1^ei C%Gf] g
D | RV.E<bf
- | , Es<Se%oYS^ - YLi sbYfCf RV.>@- ss-\C€- Enebsng
cE | RV...E s<Se%oYS^ - YLi S^ffRV.g
cc | qCz~q^ <bf†2, > <bf†† Q <bf†2RV.2<bf†iy `

```

これで同様に  $\sigma^2 R$  を求めたものが図Iiiである。濃い色が  $m^*$  の、薄い色が  $d^*$  の  $\langle \text{C@SVC S'zCqf Y} \rangle$  である。

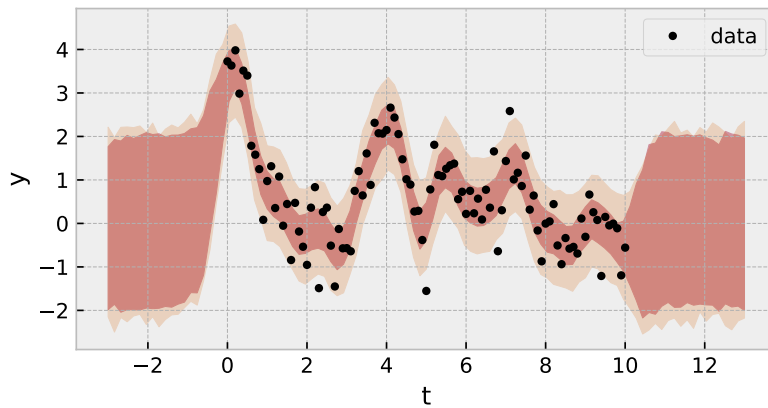


図5.5

## Ii|

さて、ここまではモデルとしては平均値がゼロのガウス過程により生成されるモデル

$$m \sim \mathcal{N}(0, \Sigma(t)) \quad \text{fiIJ} \{g$$

を考えていた。ここでは一般に平均値が  $t$  の関数となるガウス過程によりデータが生成される場合を考えよう。すなわち

$$m \sim \mathcal{N}(f(t), \Sigma(t)) \quad \text{fiIJ} g$$

の場合を考えよう。ここに  $f(t)$  は  $f(t)$  に対し、 $t = (t_0, t_1, \dots, t_{N-1})$  を要素ごとに適用したベクトル版である。ここでは  $f(t)$  として

$$f(t) = ke^{-(t-T_0)^2/2s^2} \sin(2\pi t/P) \quad \text{fiIJ} g$$

というものを考えてみよう。上では簡単に  $f(t)$  と書いたが、パラメタを  $\theta = (T_0, k, s, P)$  として  $f(t; \theta)$  という表記も併用する。

式 fiIJg の共分散行列を p3G カーネルとして選択すると、例えば、図Iivのようなデータが生成される。ここで  $\tau = 3$  であり  $f(t)$  より比較的ゆったりとしたトレンドが乗っているのが見て取れるだろう。このようなモデルを OI ; フィットするというのは、シグナル  $f(t)$  と相関のあるノイズ+観測ノイズをモデル化し、 $f(t)$  の持つパラメタを推定したいときに対応する。

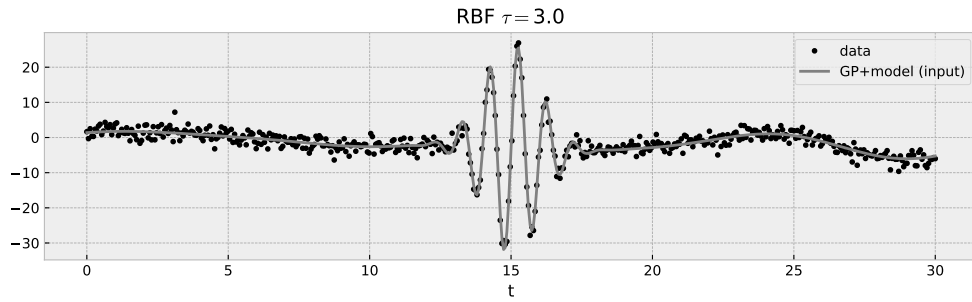


図5.6 実線は  $\sigma$  の観測ノイズなし。

$\mathbf{f}(t)$  がわかっている場合、観測ノイズを含まない、および、含んだ予測は、それぞれ、

$$p(\mathbf{m}^* | \mathbf{d}) = \mathcal{N}(\mathbf{f}(t^*) + K_{\times}^{\top} K^{-1}(\mathbf{d} - \mathbf{f}(t)), K_* - K_{\times}^{\top} K^{-1} K_{\times}) \quad \text{fl iJvg}$$

$$p(\mathbf{d}^* | \mathbf{d}) = \mathcal{N}(\mathbf{f}(t^*) + K_{\times}^{\top} K^{-1}(\mathbf{d} - \mathbf{f}(t)), K_*; -K_{\times}^{\top} K^{-1} K_{\times}) \quad \text{fl iJug}$$

となる。 $\mathbf{f}(t)$  の持つパラメタ  $(T_0, k, s, P)$  は  $O[1/k]$  ; でサンプリングされるから、例えば後者なら、サンプリングされた各  $\mathbf{t}_k$  をもちいて

$$\mathbf{d}_k^* \sim \mathcal{N}(\mathbf{f}(t^*; \mathbf{t}_k) + K_{\times}^{\top} K^{-1}(\mathbf{d} - \mathbf{f}(t; \mathbf{t}_k)), K_*; -K_{\times}^{\top} K^{-1} K_{\times}) \quad \text{fl iJDg}$$

をサンプリングすれば、予測のサンプリングができる。というわけで、以下のようにモデル  $f(t)$  込みの  $Kd$  のフィットを行うことができる。

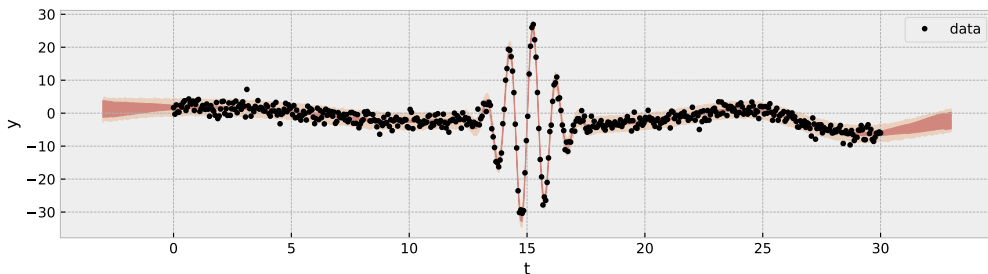


図5.7

## V

# ] ~\ d%ϕ

] ~\ d%ϕ は ddX なので様々な機能がある。この章では、] ~\ d%ϕ を使っていろいろ遊んでみよう。

## vic s<Se%T, † w] ~\ d%ϕ

s<Se%T, † w] ~\ d%ϕ により確率分布からサンプリングを試みよう。例えば、多変数正規分布の場合

```
c Hjb\ s<Se%T, † w] ~\ d%ϕ \~YzSf- qS- zC€^bq\~Y -s s\^
l Hjb\ U †i q ^@b\ S\ebqz \~YzSf- qS- zC€^bq\~Y -s U^
{
J WC%oE q- ^@b\i dp] KVC%fo]g
I ] E c|C€E
v z E ^ei YS^se- <Cf Q>I>]g
u - fC E ^ei <Cqbsf]g
D z- ~ E ci I
-
c€E zSeE€Ec
cc <bf E p3Gfz>z- ~gj zSe1^ei C%G]g
cl
cf `s<Se%o
cj @s E s\^f\C- ^E- fC ><bfE<bf > - Yyb.€sS^L~Y- q Eyq-Cgi qfsfcgi y
ci `U †
ev @UEU^fWC%o fC><bfg
eu `^~\e%ϕb
cD \^E@Szi [ ~YzSf- qS- zC] bq\~ YfYb<E- fC> <bf- qS- ^<C€\~ zqS†E<bfg
c- @^ E ^~\e%ϕbis- \eYCF o- o>\^>q^LEWC%oq- ^@b\i dp] KVC%fo] Cgg
```

こんな感じで、サンプリングできる。

ここで T, † と ] ~\ d%ϕ は @H~Y では Gd{| を用いていることに注意が必要。これは例えば zSeC€ を小さくしていくと zSeCiC€ くらいで T, † と ] ~\ d%ϕ では値が ^-^ になってしまう。

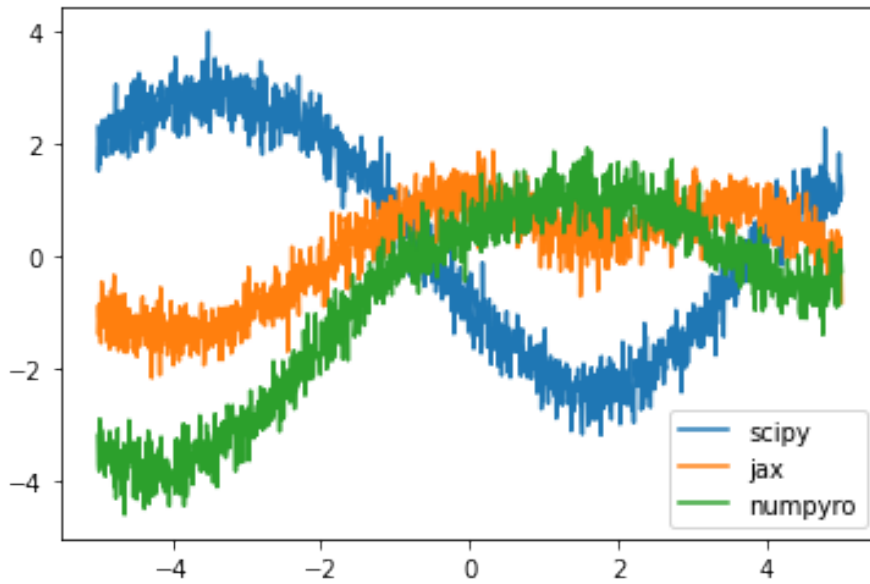


図6.1

```

c Hjb\ U #i <b^HSL S\ebqz <b^HSL
| <b^HSLi ~e@ zCf oU †€C^- 4YCE†vJo> yq-Cg

```

として GdvJ にすれば、T, †w] ~\ d%cp でも ^-^ とならない。

## WC%o

乱数の種を更新していくには U†iq ^@b\ iseYz を用いる。

```

c bWC%oq ^@b\i dp] KVC%o| Gg `bqSLS^- Y WC%o
| Hbq S S^ q ^LCf CcCg=
{ bWC%oWC%oq ^@b\i seYzfbWC%og
j @^E^- \e%qi s- \eYcf o-o> \^>q^LWC%oWC%og

```

## vicic

zq ^sHbq\

# yRdr

, ic , qfS

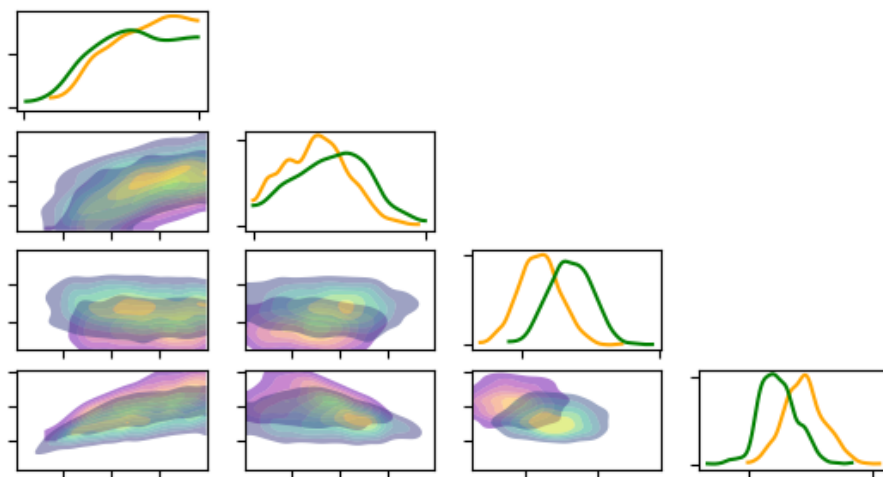
, icic

- †Gs を引き継げばよい。

```
c - †CsE- qfS<i eYbz€e- Sqfe>@SfCqLC^<CsEG YsC>\- qLS^ - YsEyq-C>sPb. EG YsCg
l - †Cs|E- qfS<i eYbz€e- Sqfe|>@SfCqLC^<CsEG YsC>\- qLS^ - YsEyq-C>sPb. EG YsC>- †E- †Csg
{ eYzi sPb. fg
```

色等を区別できるようにするためには、各種 W.:q を使い分ければ良い。例えば図, ic の設定は以下のとおりである。

```
c - †CsE- qfS<i eYbz€e- Sqfe>V$^@EoV@Co>@SfCqLC^<CsEG YsC>\- qLS^ - YsEyq-C>sPb. EG YsC>5
l V@CW.: qLsE7n<b^zb-qH€W.: qLsn=7n- YeP- n=CEI >n<\- en=neY- s\ - n8>5
{ n<b^zb-q€W.: qLsn=7n- YeP- n=88>5
j \- qLS^ - Y€W.: qLsE7n<bYbqn=nbq- ^LCn8g
i - †Cs|E- qfS<i eYbz€e- Sqfe|>V$^@EoV@Co>@SfCqLC^<CsEG YsC>\- qLS^ - YsEyq-C>sPb. EG YsC>5
v - †E- †Cs>5
u V@CW.: qLsE7n<b^zb-qH€W.: qLsn=7n- YeP- n=CEI >n<\- en=nfSqS@Ssn8>5
d n<b^zb-q€W.: qLsn=7n- YeP- n=88>5
- \- qLS^ - Y€W.: qLsE7n<bYbqn=nLqCC^ n8g
```



图A.1

3

, eeC^@f

3ic , i



9: T \ Gs 3q @4~q%pb%GqpsZL>dCqO-..W's>[ -zzPC..T \ Gs TbP^sb^>; Pq\$ XC q%? b-L- Y| -<Y~Q  
 q^>KCbqL] G~Y>, @ \ d-s:V>T-V, -^@CqYs>rW%, -^@Cq \ -^Q SYC-^@kSbŠP-^Li T, †=  
 <b\ ebs-4Yzq ^shq \ -z^s bHd%Pb^j | ~\ d%eqlq \ s>|CEI  
 9: , zS\ K~^Gs 3-%s>3-q W, i dG qX ~zzCq, Y†C%o ^@C%GfSP p-@-Y-^@TC q%q -qWt:SW@  
 , ~zb\ -z& @S Cq^zS zB^ S \ -<PSC Y q'SL= - s-qfC%o arXiv e-prints>e-LC -q† Sf=elCEiEuuv>  
 GC4q~ q%|CEI  
 9: O- \ CV-...P-q -^@VC^zb[ -s~@i 3-%s ^?%o \ S&[ -eeSLbH^ B†bQ- qP†H\ dPzb\ Cz&  
 , -q 4S&%o ApJ>\_CECg=JD-rCez\ 4Cq|CEI  
 9: p-@Hq@[ i ] G Y [ ; [ ; ~s'LO \ Sz^S ^ @%o \ Ssi arXiv e-prints>e-LC -q† Sf=elCEiCE>T~^C  
 |CEI  
 9: ? ~ dP-^>] Cq Udq @P-^>-^@[ -qS' T-^W..SW; b\ ebs-4YCB' Czs HqGY†SYC-^@, <<Cq zC@  
 dqb-4SSz& dqlq \ \ S'LS' ] ~\ d%qi arXiv e-prints>e-LC -q† Sf=elCEiEuuv>? C&\ 4Cq|CEI  
 9: pb4Cq B@..S' , C'LCqi , s\ eY-~zb\ -z& @Cf zfCCf-Y- zB^ eqlq \ i Communications of the  
 ACM>ufDgJv{ JvJ>c\_vJi