A different perspective on optimization ①

· Currently we.

. start at some point
. build a local model
. choose a new point
. report the most recent point

AND:
- see building a local model as an exercise in calculus.

· Notice:
. in this form, might be hard if:
. function evaluations are expensive
. we don't have explicit derivatives

Alternative view:

- exercise in decision making and utility

- Start with $D_0 = \{(\underline{x}_0, y_0)\}$

  value at some point

- iterate:
  - use D_i to predict $\underline{x}_{i+1}$ (a new point to eval. at)
  - form $D_{i+1} = D_i \cup \{(\underline{x}_{i+1}, y_{i+1})\}$

    value at x_{i+1}

- use D_N to predict some point — the result of the optimization

→ this could be the best point we've seen, but could be much more interesting.

Setting all this up requires some machinery. We need to manipulate uncertainty on function values. (3)

Q: I have D_i — what $\underline{x}_{i+1}$ will yield "the best" outcome?

a <u>Gaussian process</u>
is a <u>stochastic</u> <u>process</u>

generalization of the idea of a random vector to functions; ∞-dim random vector

where the mean and covariance of the values of a sample f at a <u>finite</u> set of points $\underline{x}_1 \cdots \underline{x}_i$

function made by process

is given by

$$\text{mean} = \begin{pmatrix} m(x_1) \\ m(x_2) \\ \vdots \\ m(x_i) \end{pmatrix}$$

for some fixed function m

$$\text{cov} = \begin{pmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & \\ K(x_2, x_1) & & & \\ \vdots & & & K(x_i, x_i) \end{pmatrix}$$

for some K.

<u>Notice</u>:

- K has to have special properties (at least P.D. for <u>any</u> set of points
- giving ~~ts~~ m, K yields the process

Very like a Gaussian for a super-big vector. 5

A useful trick with Gaussians

$$\begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \underline{m}_1 \\ \underline{m}_2 \end{pmatrix}, \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}\right)$$

we observe $\underline{x}_2 = \underline{a}$

Q: what is $\underline{x}_1 \mid \underline{x}_2 = \underline{a}$?

write $P = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}^{-1} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$

now

$$\log p(\underline{x}_1 \mid \underline{x}_2 = a) = \underset{\substack{\uparrow \\ \text{all terms } \underline{\text{not}} \text{ in } \underline{x}_1}}{k} - \frac{1}{2}\left[x_1^T P_{11} x_1 - 2 m_1^T P_{11} x_1 + 2 a^T P_{21} x_1 - 2 m_2^T P_{21} x_1\right]$$

Now:

- recall if $u \sim N(\mu, \Sigma)$
  then $\log p(u) = K - \frac{1}{2}(u-\mu)\Sigma^{-1}(u-\mu)$

- match terms:

  if $p(x_1 | x_2 = a)$ is $N(\mu, \Sigma)$

  - $\Sigma^{-1} = P_{11}$
  - $\mu = m_1 + P_{11}^{-1} P_{12}(a - m_2)$

Now:

$$\begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} = I$$

So: $P_{11} S_{11} = I - P_{12} S_{21}$; $P_{11} S_{12} = -P_{12} S_{22}$

So:

$$\Sigma = S_{11} - S_{12} S_{22}^{-1} S_{21}$$

$$\mu = m_1 - S_{12} S_{22}^{-1}(a - m_2).$$

All this WORKS FOR G.P.'s

for example: f is G.P., m, K

- know value at $\underline{x}_1 \cdots \underline{x}_i$ $(v_1 \cdots v_i)$
- Q: What do we know about $f(\underline{x}_r)$

some new point!

- A: it is Normal

defining property of G.P.s

- A: $f(x_r) \sim N(\mu_r, \Sigma_r)$

and we can write these out explicitly.

write $\underline{v} = \begin{pmatrix} v(x_1) \\ \vdots \\ v(x_r) \end{pmatrix}$

$$S_{11} = \begin{pmatrix} K(x_1, x_1) & & K(x_1, x_r) \\ & & \\ & & K(x_r, x_r) \end{pmatrix}$$

$$S_{21} = \begin{pmatrix} K(x_r, x_1), & \ldots & K(x_r, x_i) \end{pmatrix}$$

$$S_{12} = \begin{pmatrix} K(x_1, x_r) \\ \vdots \\ K(x_i, x_r) \end{pmatrix}$$

$$S_{22} = K(x_r, x_r)$$

$$\Sigma_r = S_{11} - S_{12} S_{22}^{-1} S_{21}$$

$$\mu_r = m(x_r) - S_{12} S_{22}^{-1} \left(v - \begin{pmatrix} m(x_1) \\ \vdots \\ m(x_i) \end{pmatrix} \right)$$

[Plug in and crank!]

a straightforward example:

$$m = 0$$
$$K(x, x') = e^{-\frac{\|x - x'\|^2}{2}}$$

Then
$$\mu_r = 0 - \sum_j (v_j) e^{-\frac{\|x_r - x_j\|^2}{2}} \quad \text{(plug in).}$$

Simple optimization strategy:

- given $x_1 \cdots x_i$, report the x_r with largest mean (so above, with some optimizer to get x_r)
- But where do $x_1 \cdots x_i$ come from? We can get them with a form of dynamic programming!
- First, more detail on representation

The expression for μ_r, Σ_r show that: (1

- w. G.P. representation, we can estimate the distribution of function values at a point. Garnett fig 2.1, 2.2 belo
p1

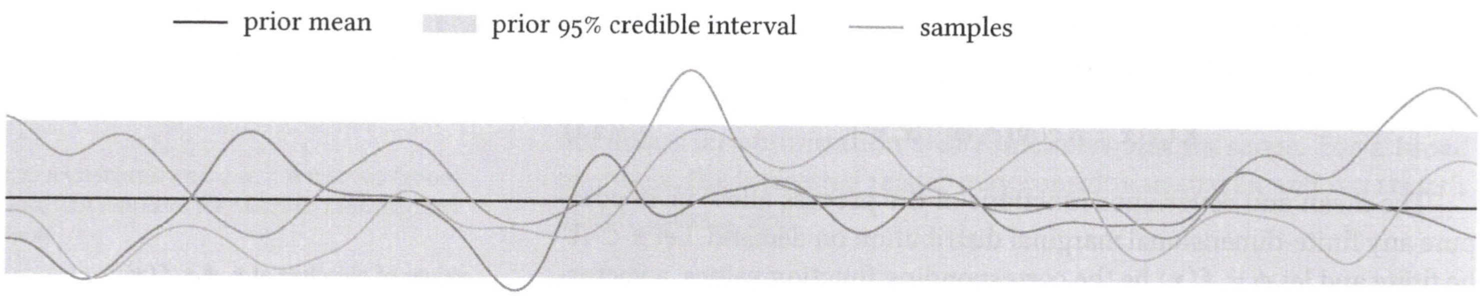


Figure 2.1: Our example Gaussian process on the domain $\mathcal{X} = [0, 30]$. We illustrate the marginal belief at every location with its mean and a 95% credible interval and also show three example functions sampled from the process.

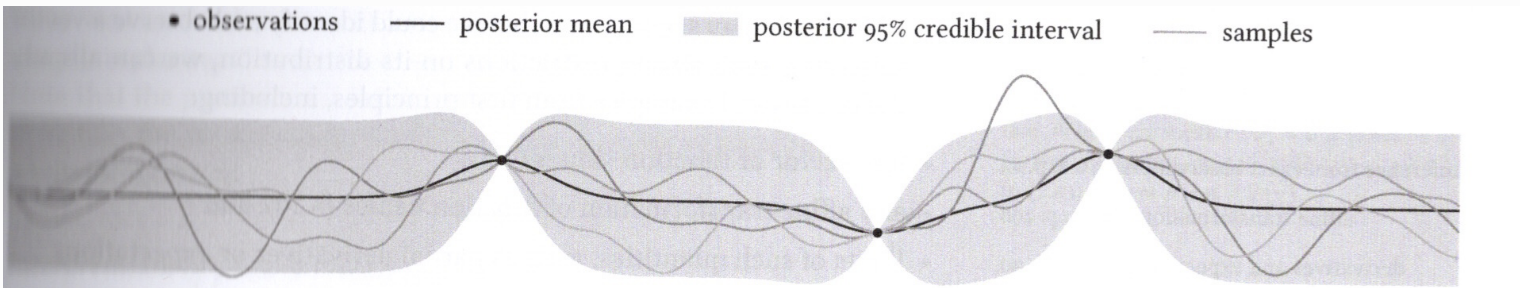


Figure 2.2: The posterior for our example scenario in Figure 2.1 conditioned on three exact observations.

Facts

- one can build a joint G.P. for more than one function (constrained or multiobjective opt)
- G.P. sample paths (the technical term for the function obtained by drawing a sample from a G.P.) are continuous if $K(x,x)$ is continuous
  - diagonal: args are =
- G.P. sample paths are differentiable if mean function is diff. AND $K(x,x)$ is diff
- There is a joint dist between function and gradient, assuming sufficiently well behaved m, K.
- for compact domain, continuous sample paths, GP sample paths have a max.
- This is unique if no two unique function values are perfectly correlated.

Common m's, K's

$m = \text{const.}$

$m = \sum_i a_i \psi_i(x).$

$K(x, x') = \exp(-d(x, x'))$

$\left[d(x, x') = \sqrt{(x-x')^T (x-x')} \right]$

$K(x, x') = \exp\left(-\frac{d^2}{2}\right)$

$K(x, x') = (1 + \sqrt{3} d) \exp(-\sqrt{3} d)$

$K(x, x') = (1 + \sqrt{5} d + \frac{5}{3} d^2) \exp(-\sqrt{5} d)$

$K(x, x') = \sum_i w_i \left[\begin{array}{l} \exp\left[-2\pi^2 (x-x')^T \Sigma_i (x-x')\right] \times \\ \cos(2\pi (x-x')^T \mu_i) \end{array} \right]$

$(w_i \geq 0)$

[There is a <u>very</u> rich theory here for K. Garnett, p 53 to start]

The <u>choice</u> is a modelling choice!

Neat trick

- imagine you want $m(x) = c$, but don't know c.
- put normal prior on c:

$$c \sim \mathcal{N}(\mu_c, \Sigma_c)$$

- f is GP conditioned on c.
- now marginalize out c.

$$f|c \sim GP(c, K).$$

so

$$f \sim GP(a, K + \Sigma_c)$$

this works for linear combination of basis functions, too (Garnett, p48)

We can now see optimization as subject to decision theory.

<u>Alg</u>:
- start w/ D. (which could be empty)

repeat:

$x = \text{policy}(D)$
(choose where to observe)

$y = \text{observe}(x)$
(observe)

$D = D \cup \{(x, y)\}$

until some termination condition

return D

report predict(D).

<u>Q</u>: predict? | <u>Many</u> <u>choices</u>
policy?

Start with simplest

$$\text{predict}(D) = \begin{bmatrix} \text{the } x \text{ w/ largest} \\ \text{function value in } D \end{bmatrix}$$

now assume we have $D_{i-1} = \left\{ \begin{matrix} (x_1, y_1) \cdots \\ (x_{i-1}, y_{i-1}) \end{matrix} \right\}$

and we get to add one point to get D_i - which point should we add?

$$\text{utility of } D_{i-1} = \cancel{\text{argmax}} \cdot \max_{y_k} (y_1 \cdots y_{i-1})$$

~~point to~~

cause that's what we'd report

we don't know function value at any other point, but we can compute expectations over function values if we pick x, then

$$\text{utility of } D_i = \text{utility of } D_{i-1} + \max(y - \hat{y}_{i-1}, 0)$$

the function value at x

the best value in D_{i-1}

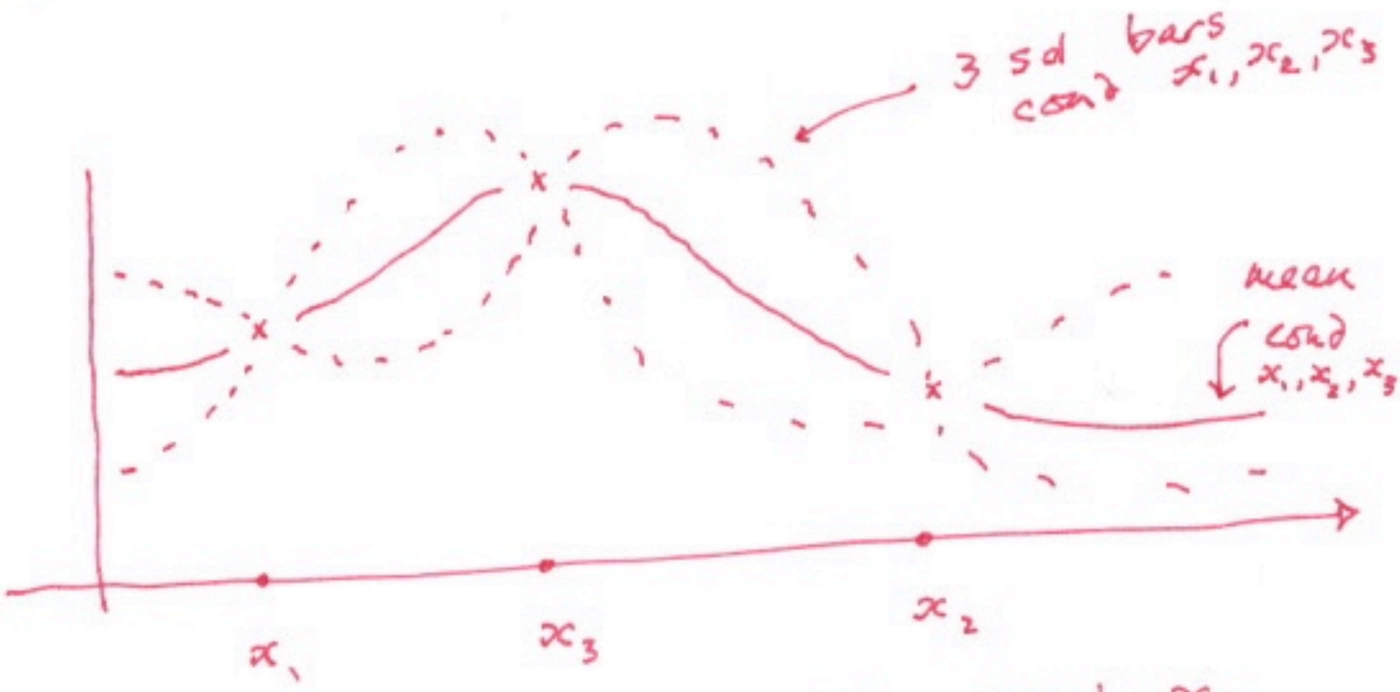
we can't compute this, but we can compute

$$E_{y|x}\left[\max(y - \hat{y}_{i-1}, 0)\right]$$

expected value of the marginal utility of choosing x

what you add to utility of D_{i-1}

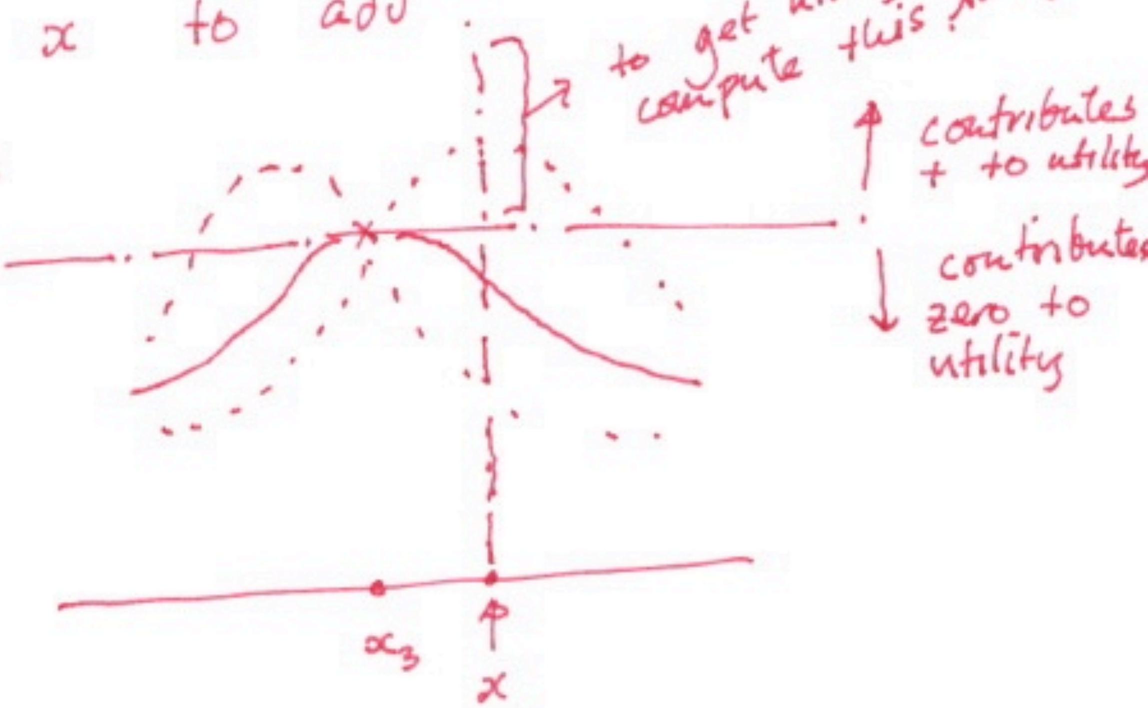
Picture :



i) if you had only D_3, would report x_3

ii) What x to add ?

iii) Zoom :



Computing the Expected marginal utility

- we have $\hat{y}_{i-1}$
- want $\frac{1}{\sqrt{2\pi}\,\sigma}\int_{-\infty}^{\infty} \max(y-\hat{y}_{i-1}, 0)\, e^{-(y-m)^2/2\sigma^2}\, dy$

where σ, m are functions of x.

- This integral is nasty, but can be evaluated
- transform $z = \frac{y-m}{\sigma}$
- to get

$$\frac{1}{\sqrt{2\pi}}\int_{\frac{\hat{y}_{i-1}-m}{\sigma}}^{\infty} \left[\sigma z - (m+\hat{y}_{i-1})\right] e^{-z^2/2}\, dz$$

- With some work, we can write in terms of error functions, etc.

- write $g_{i-1}(u) = E_{y_i | x_i = u}\left[\max(y - \hat{y}_{i-1}, 0)\right]$

- to find the best x_i, max this w.r.t. x

- Now consider D_{i-2}.
  - we want to add TWO points
  - one to get to D_{i-1} then one to get to D_i

We can find a point x_{i-1} by assuming we do the best thing with D_{i-1}

Notice, for any particular x_{i-1}

we <u>could</u>

- eval y_{i-1}
- from that, choose $x_i = \underset{u}{\operatorname{argmax}}\, g_{i-1}(u)$
- but we don't want to eval y_{i-1}
- we have $p(y_{i-1} | x_{i-1}, D_{i-2})$
- notice

  $g_{i-1}(u)$ depends on:

  y_{i-1} (because $\hat{y}_{i-1}$ might change)

  x_{i-1} (because this affects σ, μ)

- <u>write</u>:

$$g_{i-1}(u \mid y_{i-1}, x_{i-1}, D_{i-2})$$

and notice that, once we've chosen x_{i-1}, we would choose the best x_i.

· So we could <u>evaluate</u> x_{i-1} by

$$E_{y_{i-1} | x_{i-1}} \left[\max_u g(u \mid y_{i-1}, x_{i-1}, D_{i-2}) \right]$$

the expected value of x_{i-1}, assuming you then choose the best x_i.

· An induction follows <u>BUT</u> all should look unpromising

$$E\left[\max \left[E \left[\max \ \ldots \right.\right.\right.$$

should look very hard to evaluate

fairly practical strategy

- One point lookahead.
- go from D_{i-2} to D_{i-1} by
  - choosing $x_{i-1} = \underset{u}{\operatorname{argmax}}\, g_{i-2}(u; D_{i-2})$ (so ignore what you do with point in future)
  - then $x_i = \underset{u}{\operatorname{argmax}}\, g_{i-1}(u; D_{i-1})$
- Clearly, you could start at D_0
- This is a fair strategy
  - notice how it will cause x's to probe places where the function has a strong chance of being better than anything we've seen.

There is a <u>very rich</u> family of methods.

- change procedure used to make final recommendation from i
- change predictive procedure used to select next x.
- incorporate noisy function evaluations
- incorporate derivatives
- etc

<u>Examples follow:</u>

<u>Procedure used to make final rec.</u>

- have D_i $\longrightarrow$ predict what?
- we have seen predict largest observed y (simple reward)
- <u>alternative</u> predict x st $\mu(x | D_i)$ is largest (global reward)

we have an <u>expression</u> for

$$f(x_r | D_i) \sim N(\mu_r, \Sigma_r)$$

(on P7, P8)

<u>so</u>: choose x w/ largest μ_r

<u>how</u>: conventional optimization (eg Newton, etc)

<u>Alternative</u>: imagine we are <u>risk-averse</u>

- we would like ~~an~~ x w/ large y, <u>But</u> dislike too variable a reward
- we might

choose x with largest

$$\mu_r + \beta \sigma_r$$

(if β -ve, we are avoiding risk,
β +ve, " " seeking ")

<u>Choosing β is a modelling option</u>

Change decision procedure – rollout

- assume we have D_{i-2} and want to predict x_{i-1}, x_i to get D_i
- we saw
  - optimal policy (hard)
  - easier policy (not optimal)
- we could evaluate the suitability of some x to be x_{i-1} by:
  - draw sample from predictive dist for $f(x|D_{i-2})$ to get $y^?_{i-1}$
  - use this and easy policy to predict x_i
  - draw sample to get $y^?_i$
  - evaluate resulting $D^?_i$

repeat this, and average.

This is <u>rollout</u>

- how then to choose x_i ?
  - rollout estimates at a bunch of sample points
  - fit some kind of approximate function
  - choose max

One option here is a G.P. !

<u>but</u> we have noisy ests of the function values ?

G.P. with noisy observations:

- two applications
  - model rollout values
  - perhaps our function values are noisy?

- recall procedure to obtain predictive dist at $\underline{x}$, conditioned on D_i

$$\begin{matrix} y \\ \underline{o} \end{matrix} \sim N\left(\begin{matrix} m(x) \\ m(x_1) \\ \vdots \end{matrix}, \begin{matrix} K(x,x) & K(x,x_1) & \cdots \\ K(x_1,x) & & \ddots \\ \vdots & & \end{matrix} \right)$$

$$= N\left(\begin{bmatrix} \mu_r \\ \underline{m}_i \end{bmatrix}, \begin{bmatrix} K_r, & K_{ri} \\ K_{ir}, & K_{ii} \end{bmatrix} \right)$$

now condition on $\underline{o} = \underline{v}$ and use p7, p8 we now must condition on

$$\underline{O} = \underline{V} + \xi$$

know values

Noise – $N(0, \Sigma_n)$

notice noise isn't correlated to g.p.

so

$$\begin{matrix} y \\ o \end{matrix} \sim N\left(\begin{bmatrix} \mu_r \\ \underline{m}_i \end{bmatrix}, \begin{bmatrix} K_r & K_{ri} \\ K_{ir} & K_{ii} + \Sigma_n \end{bmatrix} \right)$$

and we use p7, p8 !