

A different perspective on optimization (i)

• 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 it:

- function evaluations are expensive
- we don't have explicit derivatives

Alternative view: (2)

- exercise in decision making and utility

- Start with $D_0 = \{(x_0, y_0)\}$
↑
value at some point

iterate:

- use D_i to predict

(a new point x_{i+1} to eval. at)

- form $D_{i+1} = D_i \cup \{(x_{i+1}, y_{i+1})\}$
↑
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 (3) machinery. We need to manipulate uncertainty on function values.

Q: I have D_i — what x_{i+1} will yield "the best" outcome?

a Gaussian process

is a stochastic process

↑ 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 finite set of points x_1, \dots, x_n

↑ 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

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

for some K .

Notice :

- K has to have special properties (at least P.D. for any set of points)

- giving m, K yields the process

Very like a Gaussian for a super-big $\textcircled{5}$
vector.

A useful trick with Gaussians

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

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

$$\log P(\underline{x}_1 \mid \underline{x}_2 = \underline{a}) = k - \frac{1}{2} \left[\begin{array}{l} \underline{x}_1^T P_{11} \underline{x}_1 - 2 \underline{m}_1^T P_{11} \underline{x}_1 \\ + 2 \underline{a}^T P_{21} \underline{x}_1 \\ - 2 \underline{m}_2^T P_{21} \underline{x}_1 \end{array} \right]$$

all terms not in \underline{x}_1

Now:

⑥

• 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

(7)

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

• know value at $\underline{x}_1 \dots \underline{x}_i$ ($v_1 \dots v_i$)

• Q: What do we know about $f(\underline{x}_r)$

↑
some
new
point!

• A: it is Normal
↑ defining property of G.P.'s

• A: $f(\underline{x}_r) \sim N(\mu_r, \Sigma_r)$

and we can write these out
explicitly.

8

write

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

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

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

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

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

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

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

[Plug in and crank!]

a straightforward example:

(9)

$$\mu = 0$$

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

Then

$$\mu_r = 0 - \sum_j (v_j) e^{-\frac{\|x_r - x_j\|^2}{2}}$$

(plug in).

Simple optimization strategy:

• given x_1, \dots, x_i , report the

x_r with largest mean

(so above, with some optimizer to get x_r)

- But where do x_1, \dots, x_i come from?
we can get them with a form of
dynamic programming!

- First, more detail on representation

The expressions for μ_r , Σ_r show that: (10)

- w. G.P. representation, we can estimate the distribution of function values at a point. Garnett fig 2.1, 2.2 below p 18.

Facts

(10)

- 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, G.P. sample paths have a max.
- This is unique if no two unique function values are perfectly correlated.

Common μ 's, K 's

$$\mu = \text{const.}$$

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

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

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

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

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

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

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

$$(w_i \geq 0)$$

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

The choice is a modelling choice!

Neat trick

(13)

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

$$c \sim N(\mu_c, \Sigma_c)$$

- 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. (14)

Alg: - 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 $\text{predict}(D)$.

Q: predict? | Many choices
policy?

Start with simplest

$$\text{predict}(D) = \left[\begin{array}{l} \text{the } x \text{ w/ largest} \\ \text{function value in } D \end{array} \right]$$

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

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

~~point to~~
utility of $D_{i-1} = \max_{y_i} (y_i \dots y_{i-1})$

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}

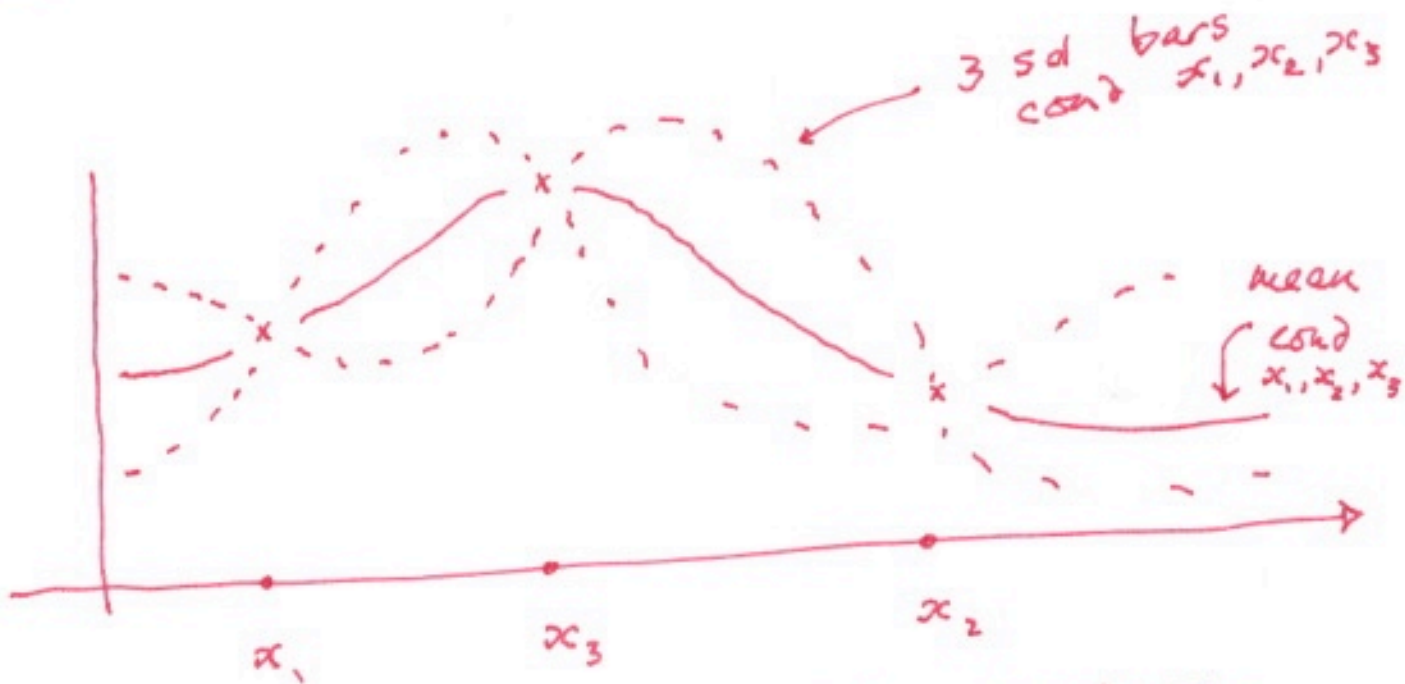
we can't compute this, but we can compute

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

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 :

