Continue : Bayesian optimization

Given a set of possible algorithms $A = \{A_1, A_2, \dots, A_n\}$

$\forall A_i \rightsquigarrow \Lambda^{(i)} \rightsquigarrow \lambda \in \Lambda^{(i)}$

AutoML : solve the opt. problem.

$$A^*_{\lambda^*} = \arg\min_{A,\lambda} \mathcal{L}(A_\lambda, D_{train}, D_{valid})$$

very expensive opt problem.

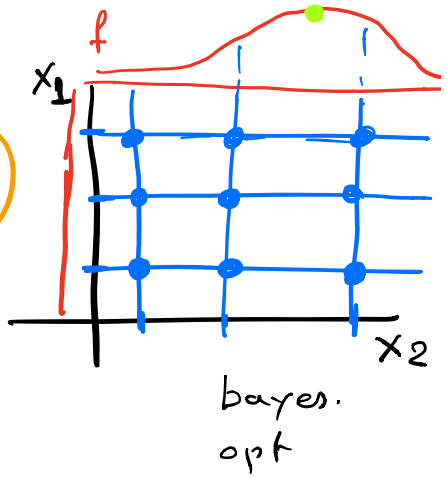
evaluating $\mathcal{L}(\cdot)$ or finding $\nabla \mathcal{L}(\cdot)$

Blackbox optimization

Discretize .. → is not fine enough (I cannot find optimal solution)

every coordinate $x_j \in \bar{X}$ and perform exhaustive search (exponential in $\bar{X}$)

grid search

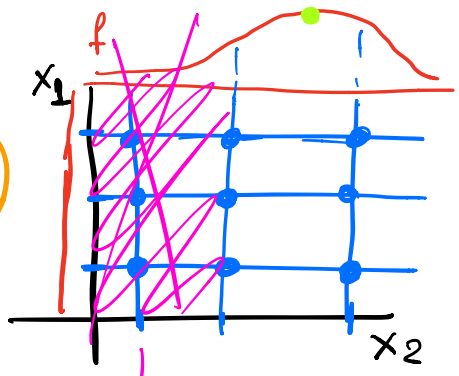


bayes. opt

$f(\bar{x})$ $\bar{X}$: some of the components are discrete; some are continuous; some are categorical.

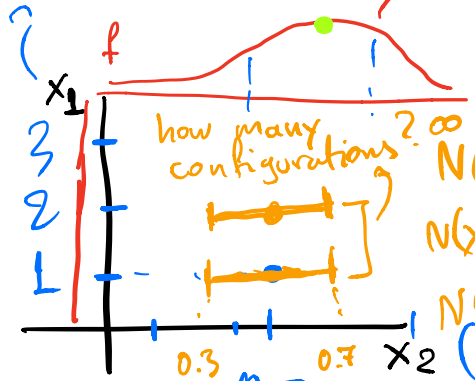
find $x^* = \arg\min_x f(x)$

Random Search



low value for f.

discrete



unknown.

how many configurations? ∞

$N(1, 0.5)$?

$N(x_1) = \{x_1 - 1, x_1 + 1, x_1\}$

$N(x_2) = [x_2 \pm \delta]$ (continuous)

Random search: avoid evaluating low-value areas

→ start from a random (x_1^1, x_2^1)

→ get the neighborhood of (x_1^1, x_2^1)

→ evaluate f only in that neighborhood

exp. and wasteful

→ take (x_1^2, x_2^2) to be the point with min f

$x_1 \in \{1, 2, 3\}$ ⇐

$x_2 \in [0, 1]$

$\delta = 0.2$

$(x_1^1, x_2^1) = (1, 0.5)$

$N = N(1, 0.5) \rightsquigarrow$? edit, reconfigure (x_1, x_2)

0.5 → 0.6 (step ?)

Known

$f(\bullet) = f(1, 0.5)$

$(2, 0.5)$
$(1, 0.6)$ x_2 is cont
$(1, 0.4)$

$N(x_1, x_2)$: a "ball" around x_1, x_2

distance in the x_1 coordinate

" " " x_2 "

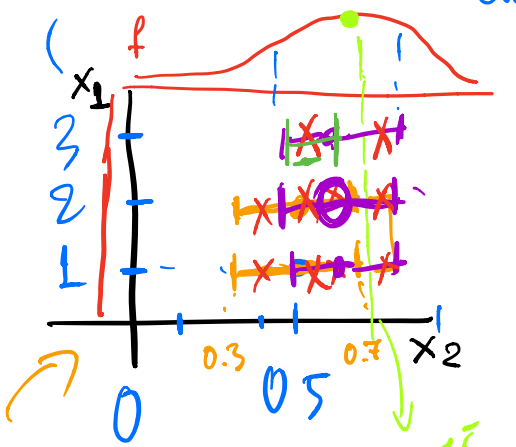
Take the product of the two sets of points that these two define.

Problem: the neighborhood $N(\bar{x})$ might require us to perform a large # of evaluations.

Attacks: (1) pick a random point in N

Ideal. multiple random points (k) ?

evaluate



→ step 2

↳ convergence is not guaranteed.

Best guarantee local optimum.

w.o. knowing anything about f

(2) → Adaptively change δ

δ → make it smaller.

Question : what is the property that we should expoit ? (monotonicity)

if f is locally monotonic wrt x_i ?

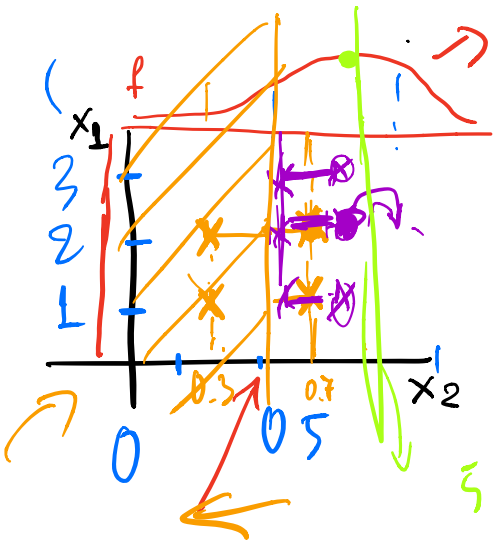
Combine Ideas

① evaluate f on a subset of points in N → ③ hope that we localy monotonic → evaluate border points.

② adaptively decreaxe δ

New idea : (exploration) pick a new random point.

helps in this scenario



What does Pruning exploit ?

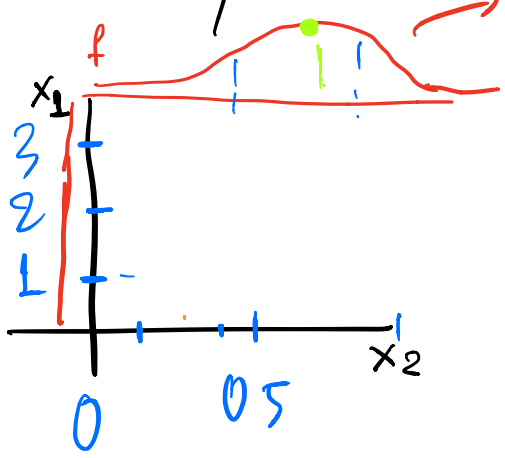
memory

Bayesian optimization

Key idea is "have memory"

we approximate f with some pdf and we will use inference to find the next best point to evaluate considering all previous evaluations.

assume a Gaussian (pdf is smooth, global opt)



what does this look like?

Gaussian.

$x^* = \arg\max_x f(x)$

$x^* = ?$ assuming it's a Gaussian?

$x^* = \mu$

Steps of BO

→ Assume a model P such that P approximates f

$$x^* = \arg\max_x P(f(x) \mid f(x_1), f(x_2), \ldots, f(x_n))$$

Posterior analysis. ① What is P?

→ fit this model (estimate P) (learning problem)

$x_1, x_2, \dots x_n$

→ $\underset{x}{\text{argmax}}\ P(\cdot)$ vs $a(P(\cdot))$

option 1 is use $P(\cdot)$ directly

option 2 is use a function of $P(\cdot)$

BO

allow us to fight local minima

acquisition → exploration

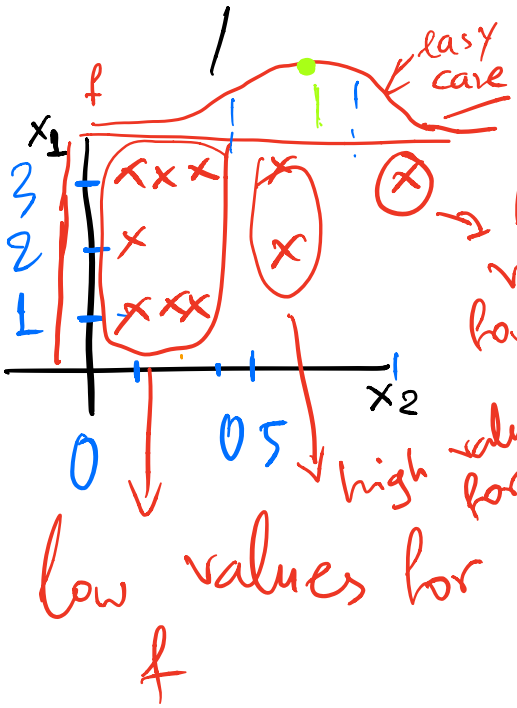
→ Consider a gaussian prior ⇒ P

→ as i am collecting evidence on the form of the real pdf I am revising my belief on P

→ I need to use an acquisition function a that takes as input my current belief on P and guides exploration of the

>pace for $\bar{x}$

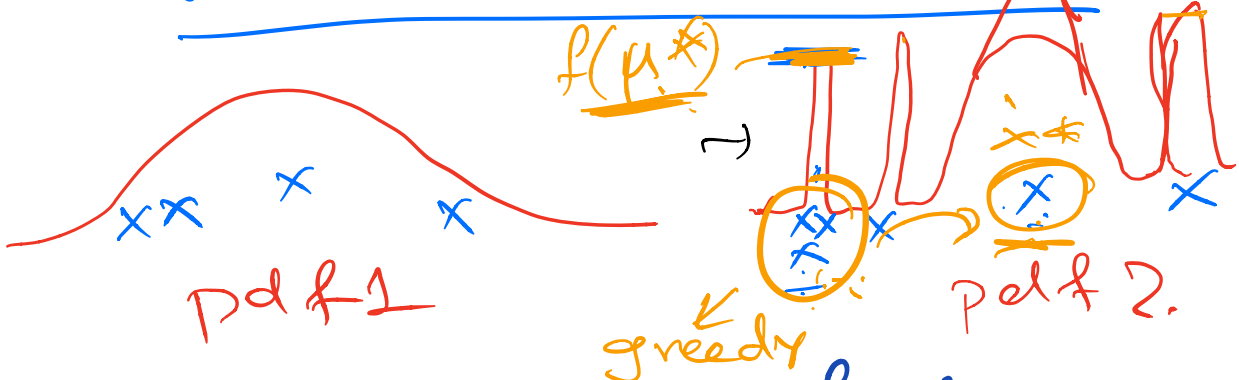
Algorithm for BO



Step 1: Take 10 samples $x_1, x_2, \ldots, x_{10}$ and evaluate f over them. (uniform)

One type of acquisition function

→ maximize exp. improvement:



Acquisition function: control exploration

→ Expected improvement. Y_{best}: running best solution

$$E\left[\min\left(f(x^*) - Y_{best}, 0\right)\right]$$

$$\frac{f(x_*) - \mu(x_*)}{\sigma(x^*)} - \frac{Y_{best} - \mu(x^*)}{\sigma(x^*)}$$

→ An acquisition function that promotes exploration

→ for different regions of X if they are not explored

confidence on what $f(x)$ is in reality is low (I have a small number of samples)

assign a positive score to these regions.

$$f(x_* \mid f(x_1)=y_1, \dots f(x_N)=y_N)$$

$$\sim N\left(k_*^T \Sigma^{-1} y,\ k(x_*,x_*) - k_*^T \Sigma^{-1} k_*\right)$$

μ : vector of evaluation.

σ^2

$$k_* = \begin{bmatrix} k(x_1,x_*) \\ k(x_2,x_*) \\ \vdots \\ k(x_N,x_*) \end{bmatrix}$$

k: kernel. update this

Σ : Covariance

$$\begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots \\ \vdots & & \\ k(x_N,x_1) & \cdots & \end{bmatrix}$$