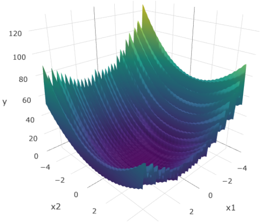
Optimization in Machine Learning

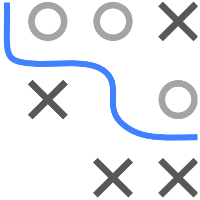
Multi-Start Optimization



Learning goals

- Multimodal functions
- Basins of Attractions
- Simple multi-start procedure

MOTIVATION



- So far: derivative-free methods for *unimodal* objective function (exception: simulated annealing)
- With multimodal objective functions, methods converge to **local minima**.
- Optimum found may differ for different starting values $\mathbf{x}^{[0]}$

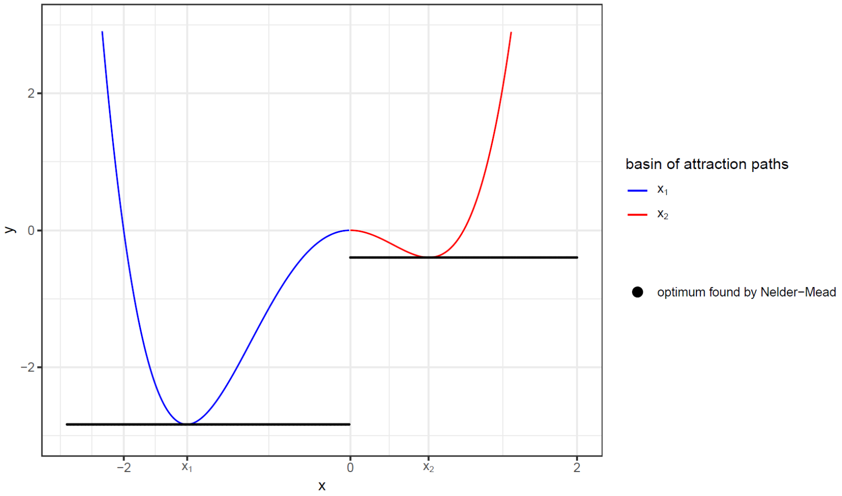
Attraction areas:

- Let $f_1^*, \dots, f_k^*$ be local minimum values of f with $f_i^* \neq f_j^* \quad \forall i \neq j$.
- Notation: $A(\mathbf{x}^{[0]})$ denotes result of algorithm A started at $\mathbf{x}^{[0]}$
- Then: Set

$$\mathcal{A}(f_i^*, A) = \{\mathbf{x} : A(\mathbf{x}) = f_i^*\}$$

is called *attraction area/basin of attraction* of f_i^* for algorithm A

ATTRACTION AREAS

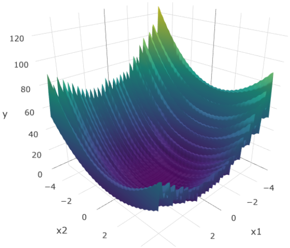


MULTI-STARTS

Levy function:

$$f(\mathbf{x}) = \sin^2(3\pi x_1) + (x_1 - 1)^2[1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + \sin^2(2\pi x_2)]$$

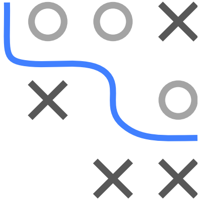
- Global minimum: $f(\mathbf{x}^*) = 0$ at $\mathbf{x}^* = (1, 1)^\top$
- Optimize f by BFGS method with random starting point in $[-2, 2]^2$ and collect result
- Repeat 100 times



Distribution of results (y values):

```
## Min.    1st Qu.   Median   Mean     3rd Qu.   Max.
## 0.0000  0.1099    0.5356   2.4351   1.9809    18.3663
```

MULTI-STARTS / 2



Idea: use multiple starting points $\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[k]}$ for algorithm A

Algorithm Multistart optimization

```
Given: optimization algorithm A(·), f : S ↦ ℝ, x ↦ f(x)
k = 0
repeat
    Draw starting point x^[k] from S (e.g. uniform if S is of finite volume)
    if k = 0 then x̂ = x^[0]
    end if
    Initialize algorithm with start value x^[k] ⇒ x̃ = A(x^[k])
    if f(x̃) < f(x̂) then x̂ = x̃
    end if
    k = k + 1
until Stop criterion fulfilled
return x̂
```

MULTI-STARTS / 3

BFGS with Multistart gives us the true minimum of the Levy function:

```
iters = 20 # number of starts
xbest = c(runif(1, -2, 2), runif(1, -2, 2))

for (i in 1:iters) {
x1 = runif(1, -2, 2)
x2 = runif(1, -2, 2)
res = optim(par = c(x1, x2), fn = f, method = "BFGS")
}

if (res$value < f(xbest)) {
xbest = res$par
}

xbest
## [1] 1 1
```