Digitise, Optimise, Visualise: Optimization

Peter H. Gruber

July 1-5, 2019

- **Root:** find (all) x such that $f(x) = 0$
- **Fixed point:** find (all) x such that $f(x) = x$
- **Inverse function:** find x such that $f(x) = y$ for a given y.
- **Approximation:** find $p(\cdot)$ such that $p(x) \approx f(x)$ around x_0
- **Minimum:** find x such that $f(x)$ is a global/local minimum.

Closely linked:

- **Interpolation:** find $f(x)$ with $x_0 < x < x_1$,
  – if only $f(x_0)$, $f(x_1)$ are known
- **Extrapolation:** find $f(x)$ with $x < x_0$ or $x > x_1$,
  – if only $f(x_0)$, $f(x_1)$ are known

Warning

- Most numerical methods rely on some property of a function.
- Inappropriate method $\longrightarrow$ possible errors.
  - May work if conditions are violated (difficult for testing)
  - Most methods always produce a result (possibly wrong)

Global vs. local properties

- *globally*, over the whole real line ($x \in \mathbb{R}$)
- *locally*,
  - within an (open or closed) interval $[x_{min}, x_{max}]$ or
  - within a neighborhood around x_0.

Before using any numerical method, verify the conditions.

Increasing

$$f(x) \leq f(x + \Delta) \quad \text{or} \quad f'(x) \geq 0.$$

For $\Delta > 0$

Decreasing

$$f(x) \geq f(x + \Delta) \quad \text{or} \quad f'(x) \leq 0.$$

Strictly inceasing/decreasing: without equal sign.

Either increasing **or** decreasing $\longrightarrow$ **monotonous**.

Alternative test for monotonicity: **horizontal line test.**
The graph of the function must cross any horizontal line only once.

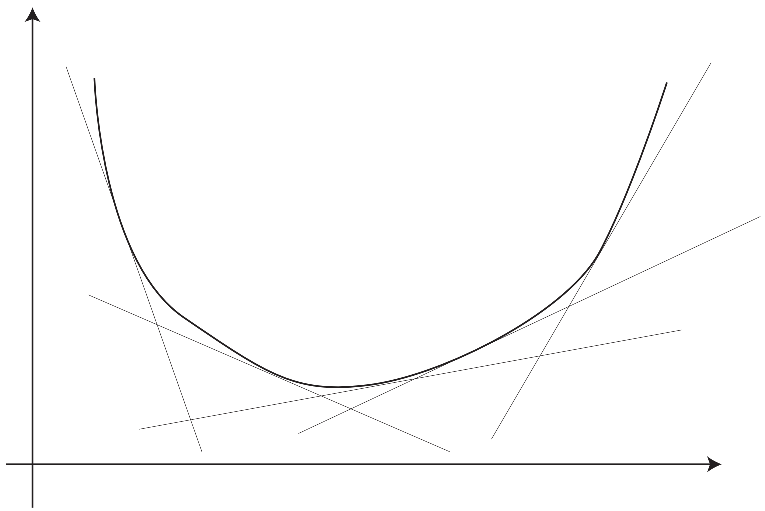
"Arbitrarily small changes in the input
$\longrightarrow$ arbitrarily small changes in the output".

Classical definition:
for every $\epsilon > 0$ there exists a $\delta > 0$ such that

$$|x - x_0| < \delta \quad \Longrightarrow \quad |f(x) - f(x_0)| < \epsilon \tag{1}$$

Alternative definition:
The graph of a continuous function can be drawn **in one strike**.

Convexity/concavity



- $\lambda f(x_1) + (1-\lambda) f(x_2) \geq f(\lambda x_1 + (1-\lambda) x_2). \quad 0 < \lambda < 1$
- A convex function is always above its tangent
- Its second derivative is positive: $f'' > 0$.
- A convex function has only one minimum.

Concave – opposite of convex.

What is a metric? A measure of **distance**.

Metric $=$ a non-negative function $d_{ij} : \mathbb{R}^n \rightarrow \mathbb{R}^+$ of two points i and j in the n-dimensional space, satisfying:

1. $d_{ii} = d_{jj} = 0$
   The distance between a point and itself is zero.
2. if $i \neq j \rightarrow d_{ij} > 0$
   The distance between two different points is larger than zero.
3. $d_{ij} = d_{ji}$
   The distance from i to j as the same as the distance from j to i
4. $d_{ik} \leq d_{ij} + d_{jk}$
   The triangle inequality is fulfilled.

Optimization

- **Minimze** cost, error, damage, risk ...
- **Optimize** solution, use of resources, portfolio ...
- **Maximize** output, profit, likelihood, ...
- **M-estimation** huge class of estimators in econometrics

Sufficient to discuss minimization:

- Optimum is always a maximum or minimum.
- max $f(x) =$ min $-f(x)$

A static **minimization problem** is a problem of the type

$$\min_{\theta} f(\theta; X) \qquad f : \mathbb{R}^n \to \mathbb{R}, \quad \theta \in \Theta \subset \mathbb{R}^n \tag{2}$$

$f(\cdot)$ . . . *objective function* $\qquad$ Θ . . . *feasible set*.

First and second derivative.
If $f(\cdot)$ is twice differentiable, a (local) minimum is found where

$$
\begin{array}{llll}
\text{1-dim:} & f'(x) = 0 & \text{and} & f''(x) > 0 \\
n\text{-dim:} & \frac{\partial f}{\partial \mathbf{x}} = 0 & \text{and} & \text{Hessian } \frac{\partial^2 f}{\partial x_i \partial x_j} \text{ pos. definite}
\end{array}
$$

Optimization problem $\longleftrightarrow$ find the root of the first derivative.

Existence of minima
Weierstrass extreme value theorem: any continuous $f(\cdot)$ on a closed interval has a minimum and a maximum. (Sometimes at border)

Minimum preserving transformations
– Any affine transformation $y_{new} = \alpha + \beta \cdot x$ (with $\beta > 0$)
– Generally every strictly increasing function (i.e. any power on $\mathbb{R}^+$)
Application: simplify calculations.

Classification of minimization algorithms

Every algorithm uses some property of the objective function
$\rightarrow$ need *some* knowledge of the problem.

Convexity
- Most important distinction. Is objective fn convex?

Use of derivative
- explicitly (functional form of the derivative known)
- implicitly (derivative calculated numerically, but must exist)

Starting values
- One starting point/vector (most algorithms)
- Starting interval (grid search, genetic algorithms)

Type of algorithm
- Deterministic or stochastic

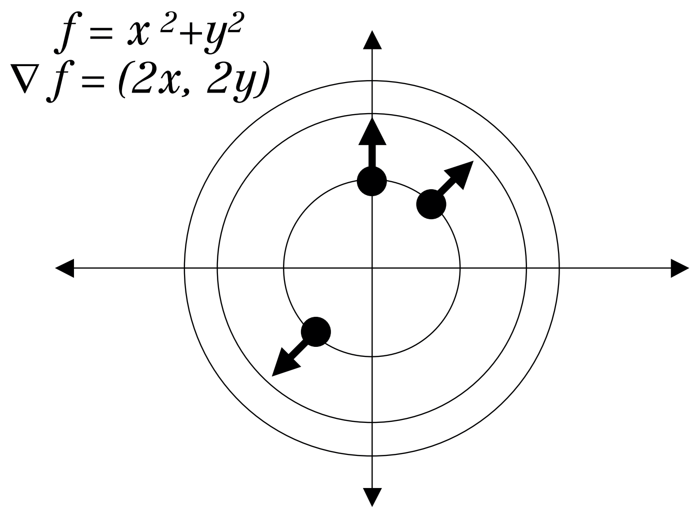
Two-step algorithms combine some of the above features.

Convex optimization

- Idea: walk uphill to reach the mountain top.
- Direction of the steepest incline = *gradient* ∇f
- Steepest descent $= -\nabla f$.

The gradient of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the vector of its partial derivatives

$$grad\ f(x_1, x_2) = \frac{\partial f}{\partial \mathbf{x}} = \nabla f = \begin{pmatrix} \frac{\partial f}{x_1} \\ \frac{\partial f}{x_2} \end{pmatrix}$$



Gradient-based methods: steepest descent 2

1. Starting point x_0, calculate gradient $\nabla f(x_0)$
2. **Direction:** $\nabla f(x_n)$
   Distance: s
   Iteration: $x_{n+1} = x_n - s\nabla f(x_n)$.
   Optimal value for s:
   (a) Bracketing: start with arbitrary interval for $s \in [a, b]$. Expand interval $(a/2, \dots)$ and $(2b, \dots)$ until $f(x_n - s\nabla f(x_n))$ gets *worse*.
   (b) Minimization: use a bisection find minimizing s
3. Repeat (2) until $||f(x_{n+1}) - f(x_n)|| < \epsilon_{target}$

   Note: $\nabla f(x_n) \approx 0$ near a minimum/maximum

Gradient-based methods: steepest descent 3

Steepest descent

Convex, derivative-based, deterministic, parallel up to # of variables

Advantages	**Disadvantages**
+ Very fast	– Explicit gradient rarely available
+ Intuitive algorithm	– Numerical gradient costly/instalbe
	– Cannot handle discontinuities

+/– First order approximation: fast, simple, not precise

Advanced gradient-based methods

– Using second derivatives (Hessian matrix)
– Allowing for larger approximation region
– Quasi-Newton methods to approximate Hessian matrix