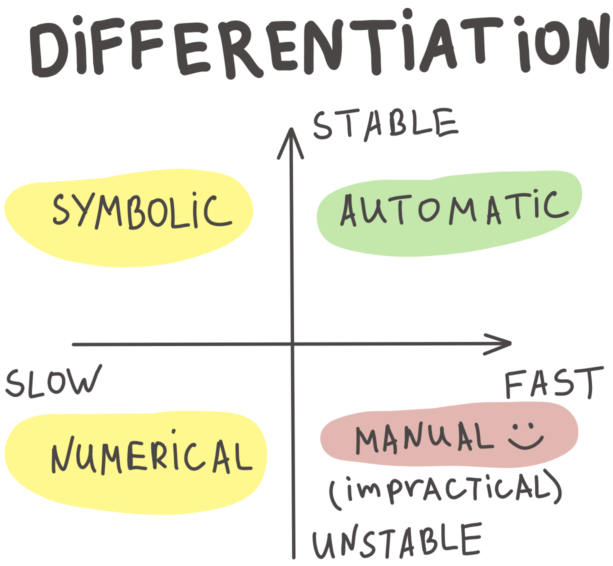
Idea



Automatic differentiation is a scheme, that allows you to compute a value of gradient of function with a cost of computing function itself only twice.

Chain rule

We will illustrate some important matrix calculus facts for specific cases

Univariate chain rule

Suppose, we have the following functions $R : \mathbb{R} \to \mathbb{R}, L : \mathbb{R} \to \mathbb{R}$ and $W \in \mathbb{R}$. Then

$$
\frac{\partial R}{\partial W} = \frac{\partial R}{\partial L} \frac{\partial L}{\partial W}
$$

Multivariate chain rule

The simplest example:

$$
\frac{\partial}{\partial t} f(x_1(t), x_2(t)) = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}
$$

Now, we'll consider $f : \mathbb{R}^n \to \mathbb{R}$:

$$
\frac{\partial}{\partial t} f(x_1(t), \ldots, x_n(t)) = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \ldots + \frac{\partial f}{\partial x_n} \frac{\partial x_n}{\partial t}
$$

But if we will add another dimension $f : \mathbb{R}^n \to \mathbb{R}^m$, than the j-th output of f will be:

$$\frac{\partial}{\partial t} f_j(x_1(t), \ldots, x_n(t)) = \sum_{i=1}^{n} \frac{\partial f_j}{\partial x_i} \frac{\partial x_i}{\partial t} = \sum_{i=1}^{n} J_{ji} \frac{\partial x_i}{\partial t},$$

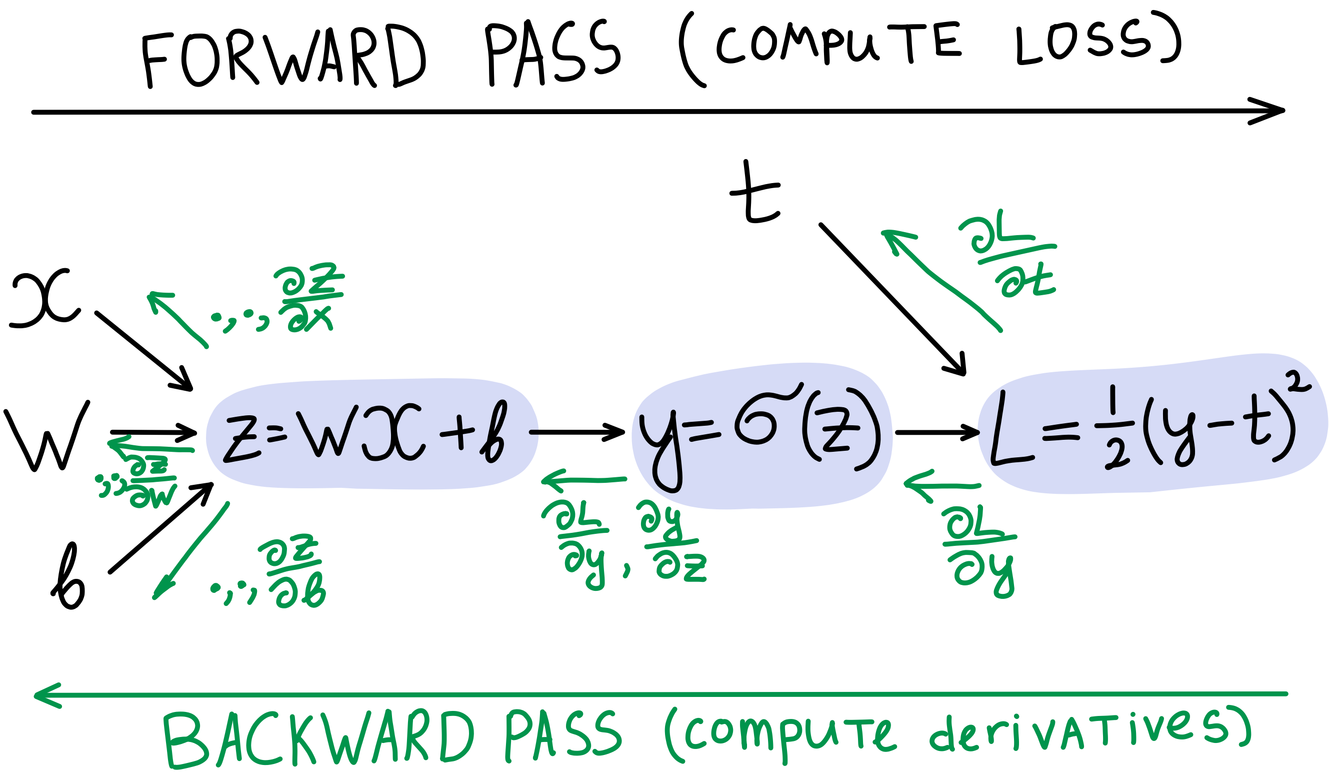
where matrix $J \in \mathbb{R}^{m \times n}$ is the jacobian of the f. Hence, we could write it in a vector way:

$$\frac{\partial f}{\partial t} = J \frac{\partial x}{\partial t} \quad \Longleftrightarrow \quad \left(\frac{\partial f}{\partial t}\right)^\top = \left(\frac{\partial x}{\partial t}\right)^\top J^\top$$

Backpropagation

The whole idea came from the applying chain rule to the computation graph of primitive operations

$$L = L\left(y\left(z(w, x, b)\right), t\right)$$



$$\begin{aligned}
z &= wx + b & \frac{\partial z}{\partial w} = x, \frac{\partial z}{\partial x} = w, \frac{\partial z}{\partial b} = 0 \\
y &= \sigma(z) & \frac{\partial y}{\partial z} = \sigma'(z) \\
L &= \frac{1}{2}(y - t)^2 & \frac{\partial L}{\partial y} = y - t, \frac{\partial L}{\partial t} = t - y
\end{aligned}$$

All frameworks for automatic differentiation construct (implicitly or explicitly) computation graph. In deep learning we typically want to compute the derivatives of the loss function L w.r.t. each intermediate parameters in order to tune them via gradient descent. For this purpose it is convenient to use the following notation:

$$\overline{v_i} = \frac{\partial L}{\partial v_i}$$

Let $v_1, \ldots, v_N$ be a topological ordering of the computation graph (i.e. parents come before children). v_N denotes the variable we're trying to compute derivatives of (e.g. loss).

Forward pass:

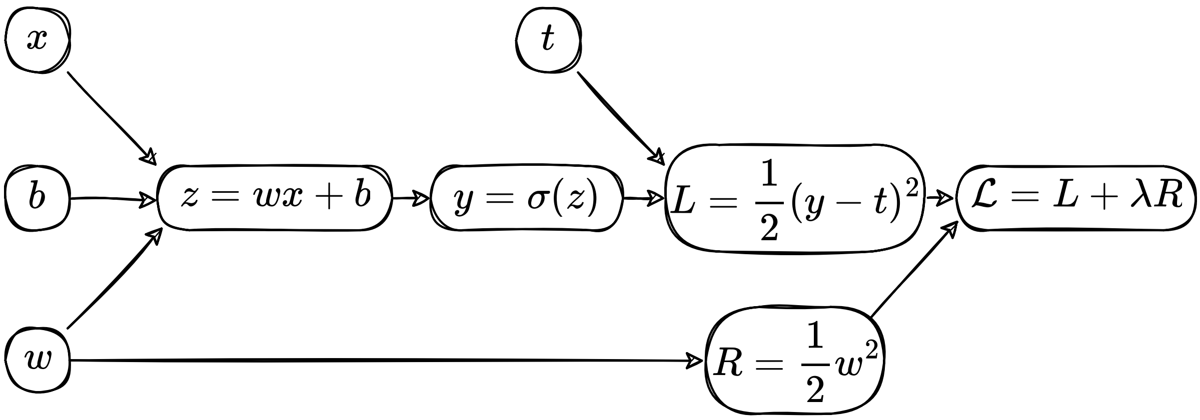
- For $i = 1, \ldots, N$:
  - Compute v_i as a function of its parents.

Backward pass:

- $\overline{v_N} = 1$
- For $i = N-1, \ldots, 1$:
  - Compute derivatives $\overline{v_i} = \sum_{j \in \mathrm{Children}(v_i)} \overline{v_j} \frac{\partial v_j}{\partial v_i}$

Note, that $\overline{v_j}$ term is coming from the children of $\overline{v_i}$, while $\frac{\partial v_j}{\partial v_i}$ is already precomputed effectively.

Univariate logistic least squares regression



Forward pass

$$
\begin{aligned}
z &= wx + b \\
y &= \sigma(z) \\
L &= \frac{1}{2}(y-t)^2 \\
R &= \frac{1}{2}w^2 \\
\mathcal{L} &= L + \lambda R
\end{aligned}
$$

Backward pass

$$
\begin{aligned}
\overline{\mathcal{L}} &= 1 \\
\overline{R} &= \overline{\mathcal{L}}\frac{d\mathcal{L}}{dR} = \overline{\mathcal{L}}\lambda \\
\overline{L} &= \overline{\mathcal{L}}\frac{d\mathcal{L}}{dL} = \overline{\mathcal{L}} \\
\overline{y} &= \overline{L}\frac{dL}{dy} = \overline{L}(y-t)
\end{aligned}
$$

$$
\begin{aligned}
\overline{z} &= \overline{y}\frac{dy}{dz} = \overline{y}\sigma'(z) \\
\overline{w} &= \overline{z}\frac{dz}{dw} + \overline{R}\frac{dR}{dw} = \overline{z}x + \overline{R}w \\
\overline{b} &= \overline{z}\frac{dz}{db} = \overline{z} \\
\overline{x} &= \overline{z}\frac{dz}{dx} = \overline{z}w
\end{aligned}
$$

Jacobian vector product

The reason why it works so fast in practice is that the Jacobian of the operations are already developed in effective manner in automatic differentiation frameworks. Typically, we even do not construct or store the full Jacobian, doing matvec directly instead.

Example: element-wise exponent

$$
y = \exp(z) \qquad J = \text{diag}(\exp(z)) \qquad \overline{z} = \overline{y}J
$$

See the examples of Vector-Jacobian Products from autodidact library:

```
defvjp(anp.add,         lambda g, ans, x, y : unbroadcast(x, g),
                        lambda g, ans, x, y : unbroadcast(y, g))
defvjp(anp.multiply,    lambda g, ans, x, y : unbroadcast(x, y * g),
                        lambda g, ans, x, y : unbroadcast(y, x * g))
defvjp(anp.subtract,    lambda g, ans, x, y : unbroadcast(x, g),
                        lambda g, ans, x, y : unbroadcast(y, -g))
defvjp(anp.divide,      lambda g, ans, x, y : unbroadcast(x,   g / y),
                        lambda g, ans, x, y : unbroadcast(y, - g * x / y**2))
defvjp(anp.true_divide, lambda g, ans, x, y : unbroadcast(x,   g / y),
```

```
                                 lambda g, ans, x, y : unbroadcast(y, - g * x / y**2))
```

Hessian vector product

Interesting, that the similar idea could be used to compute Hessian-vector products, which is essential for second order optimization or conjugate gradient methods. For a scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ with continuous second derivatives (so that the Hessian matrix is symmetric), the Hessian at a point $x \in \mathbb{R}^n$ is written as $\partial^2 f(x)$. A Hessian-vector product function is then able to evaluate

$$v \mapsto \partial^2 f(x) \cdot v$$

for any vector $v \in \mathbb{R}^n$.

The trick is not to instantiate the full Hessian matrix: if n is large, perhaps in the millions or billions in the context of neural networks, then that might be impossible to store. Luckily, `grad` (in the jax/autograd/pytorch/tensorflow) already gives us a way to write an efficient Hessian-vector product function. We just have to use the identity

$$\partial^2 f(x) v = \partial [x \mapsto \partial f(x) \cdot v] = \partial g(x),$$

where $g(x) = \partial f(x) \cdot v$ is a new vector-valued function that dots the gradient of f at x with the vector v. Notice that we're only ever differentiating scalar-valued functions of vector-valued arguments, which is exactly where we know `grad` is efficient.

```python
import jax.numpy as jnp

def hvp(f, x, v):
    return grad(lambda x: jnp.vdot(grad(f)(x), v))(x)
```

Code

Open in Colab

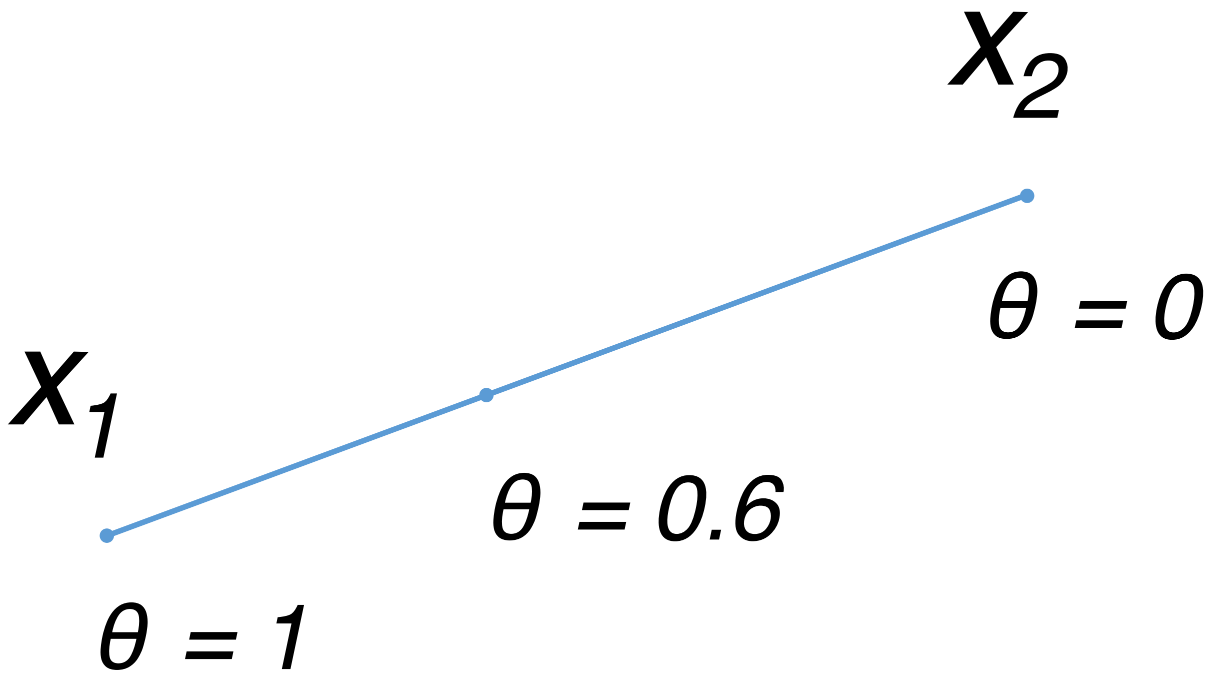
Materials

- Autodidact - a pedagogical implementation of Autograd
- CSC321 Lecture 6
- CSC321 Lecture 10
- Why you should understand backpropagation :)
- JAX autodiff cookbook
- Materials from CS207: Systems Development for Computational Science course with very intuitive explanation.

Line segment

Suppose x_1, x_2 are two points in $\mathbb{R}^n$. Then the line segment between them is defined as follows:

$$x = \theta x_1 + (1-\theta)x_2, \ \theta \in [0,1]$$



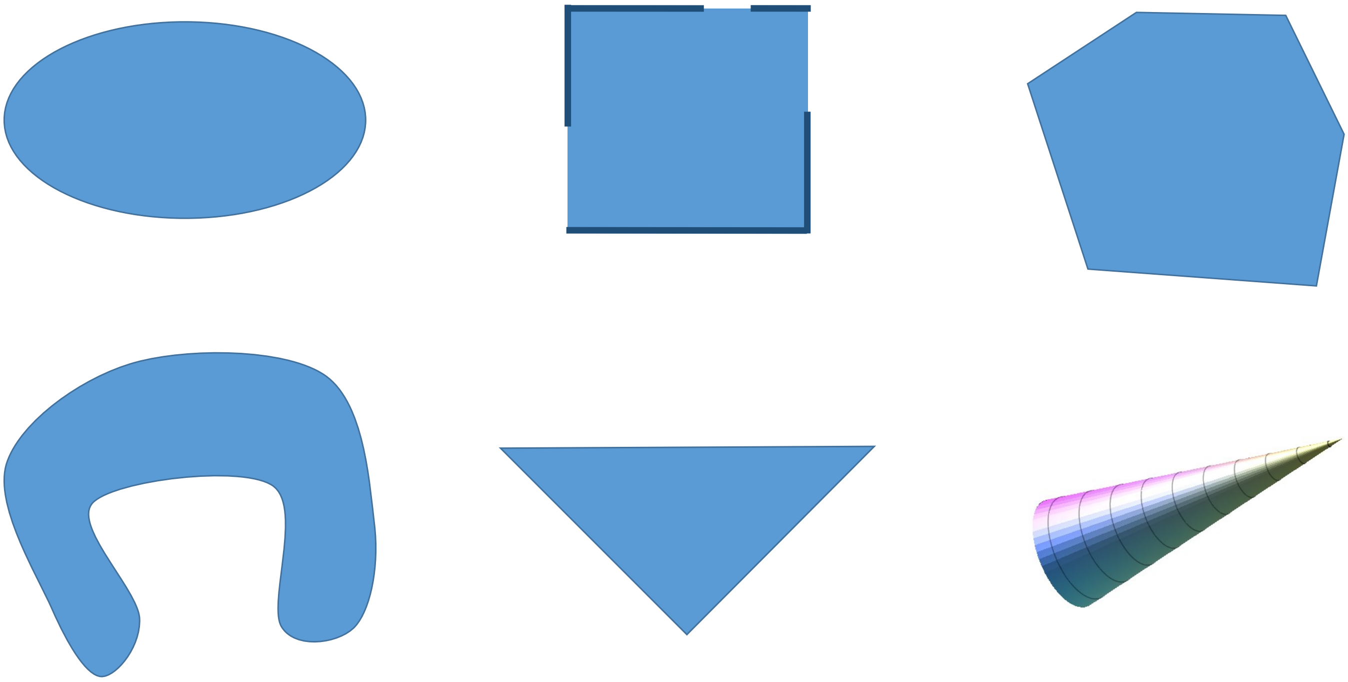
Convex set

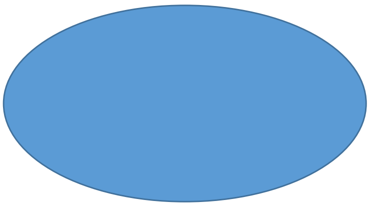
The set S is called **convex** if for any x_1, x_2 from S the line segment between them also lies in S, i.e.

$$\forall \theta \in [0,1], \ \forall x_1, x_2 \in S : \theta x_1 + (1-\theta)x_2 \in S$$

Examples:

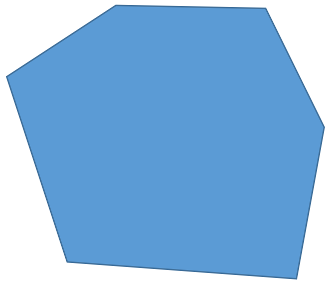
- Any affine set
- Ray
- Line segment





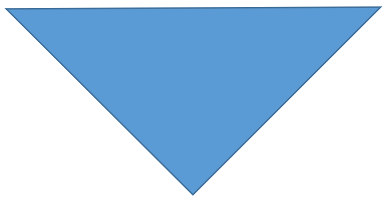
BRO

NOT BRO

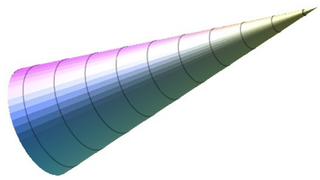


BRO

NOT BRO



BRO



BRO

Related definitions

Convex combination

Let $x_1, x_2, \dots, x_k \in S$, then the point $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ is called the convex combination of points $x_1, x_2, \dots, x_k$ if $\sum\limits_{i=1}^{k} \theta_i = 1, \ \theta_i \geq 0.$

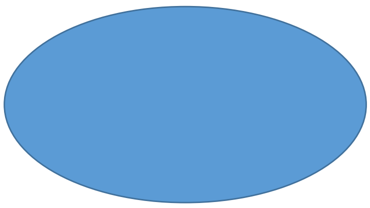
Convex hull

The set of all convex combinations of points from S is called the convex hull of the set S.

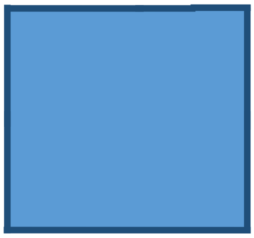
$$
\mathbf{conv}(S) = \left\{ \sum_{i=1}^{k} \theta_i x_i \mid x_i \in S, \sum_{i=1}^{k} \theta_i = 1, \; \theta_i \geq 0 \right\}
$$

- The set $\mathbf{conv}(S)$ is the smallest convex set containing S.
- The set S is convex if and only if $S = \mathbf{conv}(S)$.

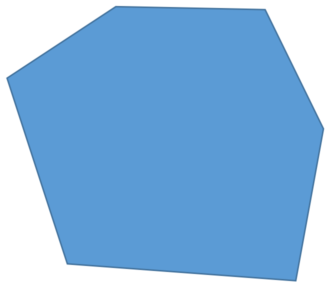
Examples:



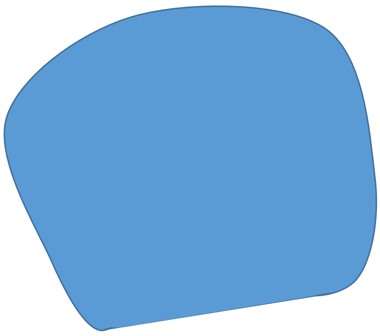
BRO



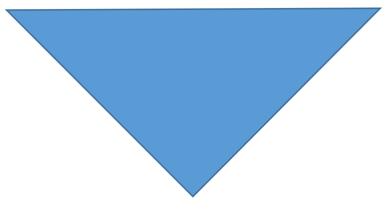
BRO



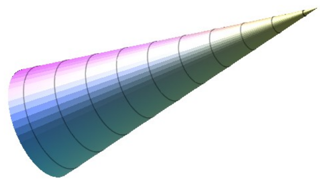
BRO



BRO



BRO



BRO

Finding convexity

In practice it is very important to understand whether a specific set is convex or not. Two approaches are used for this depending on the context.

- By definition.
- Show that S is derived from simple convex sets using operations that preserve convexity.

By definition

$$x_1, x_2 \in S, \ 0 \leq \theta \leq 1 \quad \rightarrow \quad \theta x_1 + (1-\theta)x_2 \in S$$

Preserving convexity

The linear combination of convex sets is convex

Let there be 2 convex sets S_x, S_y, let the set
$S = \{s \mid s = c_1 x + c_2 y, \ x \in S_x, \ y \in S_y, \ c_1, c_2 \in \mathbb{R}\}$

Take two points from S: $s_1 = c_1 x_1 + c_2 y_1, s_2 = c_1 x_2 + c_2 y_2$ and prove that the segment between them $\theta s_1 + (1-\theta)s_2, \theta \in [0,1]$ also belongs to S

$$\theta s_1 + (1-\theta)s_2$$

$$\theta(c_1 x_1 + c_2 y_1) + (1-\theta)(c_1 x_2 + c_2 y_2)$$

$$c_1(\theta x_1 + (1-\theta)x_2) + c_2(\theta y_1 + (1-\theta)y_2)$$

$$c_1 x + c_2 y \in S$$

The intersection of any (!) number of convex sets is convex

If the desired intersection is empty or contains one point, the property is proved by definition. Otherwise, take 2 points and a segment between them. These points must lie in all intersecting sets, and since they are all convex, the segment between them lies in all sets and, therefore, in their intersection.

The image of the convex set under affine mapping is convex

$$S \subseteq \mathbb{R}^n \text{ convex} \quad \rightarrow \quad f(S) = \{f(x) \mid x \in S\} \text{ convex} \qquad (f(x) = \mathbf{A}x + \mathbf{b})$$

Examples of affine functions: extension, projection, transposition, set of solutions of linear matrix inequality $\{x \mid x_1 A_1 + \ldots + x_m A_m \preceq B\}$. Here $A_i, B \in \mathbf{S}^p$ are symmetric matrices $p \times p$.

Note also that the prototype of the convex set under affine mapping is also convex.

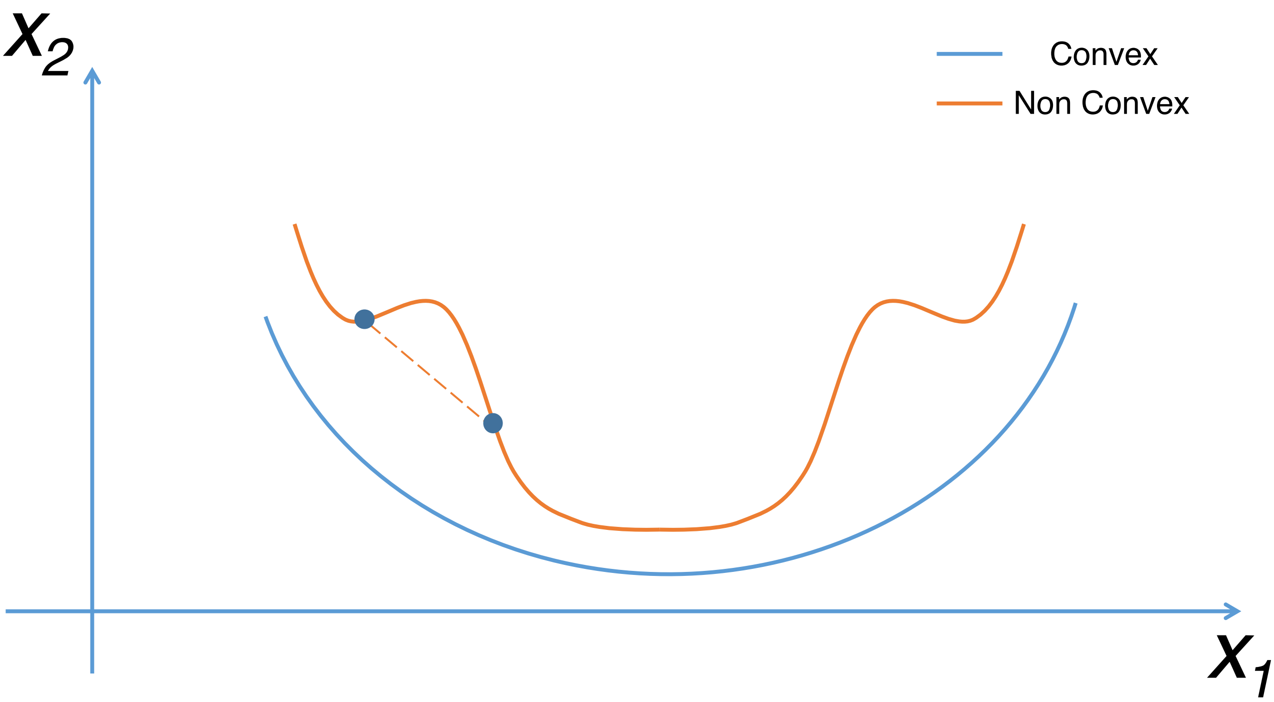
$$S \subseteq \mathbb{R}^m \text{ convex} \ \rightarrow \ f^{-1}(S) = \{x \in \mathbb{R}^n \mid f(x) \in S\} \text{ convex} \quad (f(x) = \mathbf{A}x + \mathbf{b})$$

Convex function

The function $f(x)$, **which is defined on the convex set $S \subseteq \mathbb{R}^n$**, is called **convex** on S, if:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

for any $x_1, x_2 \in S$ and $0 \leq \lambda \leq 1$.
If above inequality holds as strict inequality $x_1 \neq x_2$ and $0 < \lambda < 1$, then function is called strictly convex on S.



Examples

- $f(x) = x^p, \quad p > 1, \quad x \in \mathbb{R}_+$
- $f(x) = \|x\|^p, \quad p > 1, x \in \mathbb{R}^n$
- $f(x) = e^{cx}, \quad c \in \mathbb{R}, x \in \mathbb{R}$
- $f(x) = -\ln x, \quad x \in \mathbb{R}_{++}$
- $f(x) = x \ln x, \quad x \in \mathbb{R}_{++}$
- The sum of the largest k coordinates $f(x) = x_{(1)} + \ldots + x_{(k)}, \quad x \in \mathbb{R}^n$
- $f(X) = \lambda_{max}(X), \quad X = X^T$

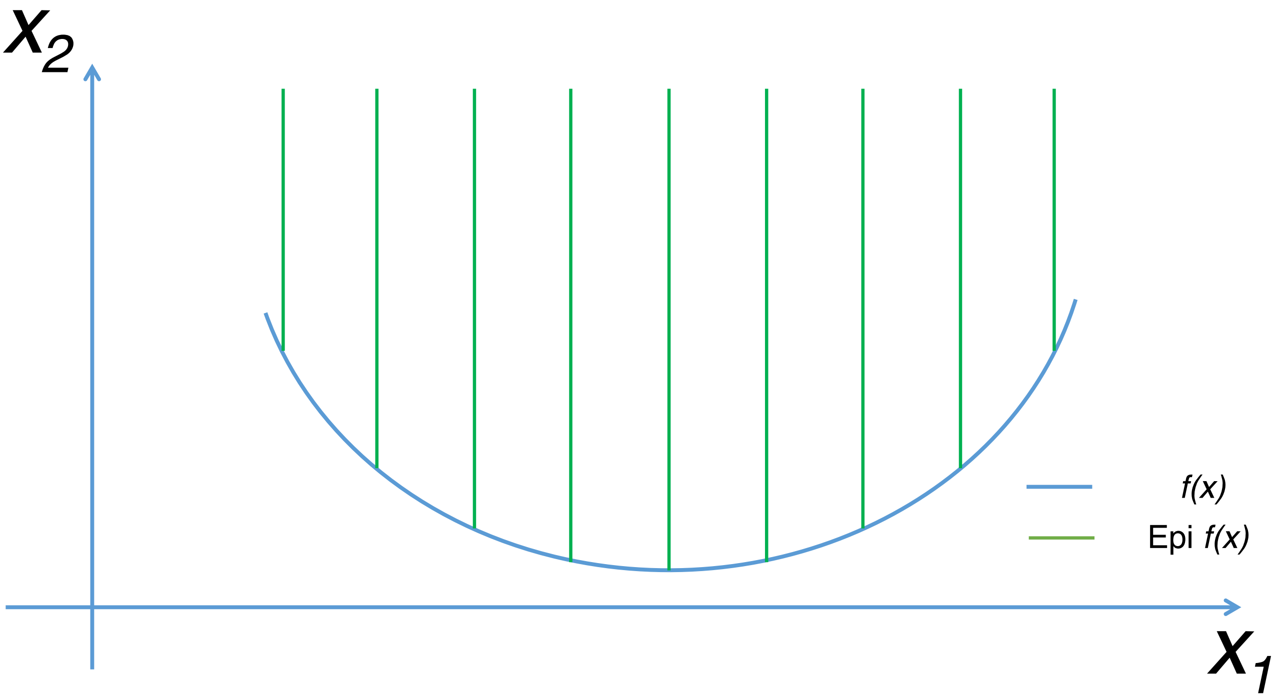
- $f(X) = -\log \det X, \quad X \in S^n_{++}$

Epigraph

For the function $f(x)$, defined on $S \subseteq \mathbb{R}^n$, the following set:

$$\text{epi } f = \{[x, \mu] \in S \times \mathbb{R} : f(x) \leq \mu\}$$

is called **epigraph** of the function $f(x)$.

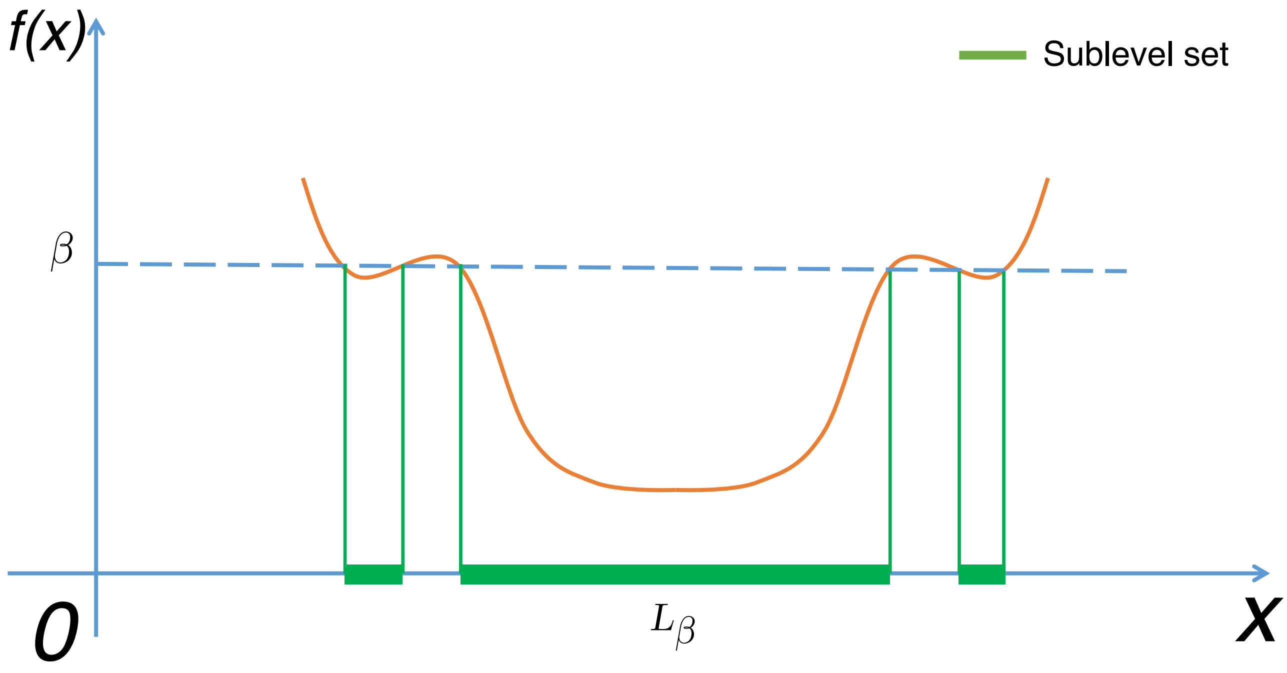


Sublevel set

For the function $f(x)$, defined on $S \subseteq \mathbb{R}^n$, the following set:

$$\mathcal{L}_\beta = \{x \in S : f(x) \leq \beta\}$$

is called **sublevel set** or Lebesgue set of the function $f(x)$.



Criteria of convexity

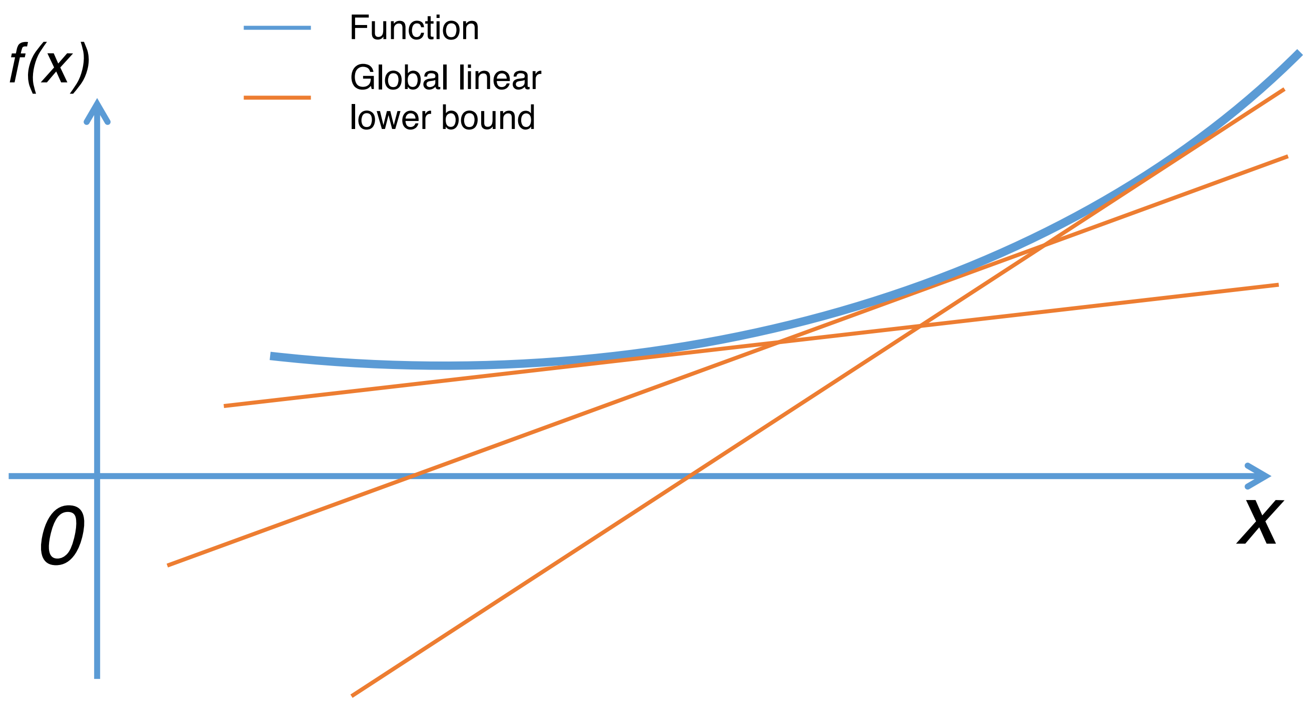
First order differential criterion of convexity

The differentiable function $f(x)$ defined on the convex set $S \subseteq \mathbb{R}^n$ is convex if and only if $\forall x, y \in S$:

$$
f(y) \geq f(x) + \nabla f^T(x)(y - x)
$$

Let $y = x + \Delta x$, then the criterion will become more tractable:

$$
f(x + \Delta x) \geq f(x) + \nabla f^T(x)\Delta x
$$



Second order differential criterion of convexity

Twice differentiable function $f(x)$ defined on the convex set $S \subseteq \mathbb{R}^n$ is convex if and only if $\forall x \in \mathbf{int}(S) \neq \emptyset$:

$$\nabla^2 f(x) \succeq 0$$

In other words, $\forall y \in \mathbb{R}^n$:

$$\langle y, \nabla^2 f(x) y \rangle \geq 0$$

Connection with epigraph

The function is convex if and only if its epigraph is a convex set.

Connection with sublevel set

If $f(x)$ - is a convex function defined on the convex set $S \subseteq \mathbb{R}^n$, then for any β sublevel set $\mathcal{L}_\beta$ is convex.

The function $f(x)$ defined on the convex set $S \subseteq \mathbb{R}^n$ is closed if and only if for any β sublevel set $\mathcal{L}_\beta$ is closed.

Reduction to a line

$f : S \to \mathbb{R}$ is convex if and only if S is a convex set and the function $g(t) = f(x + tv)$

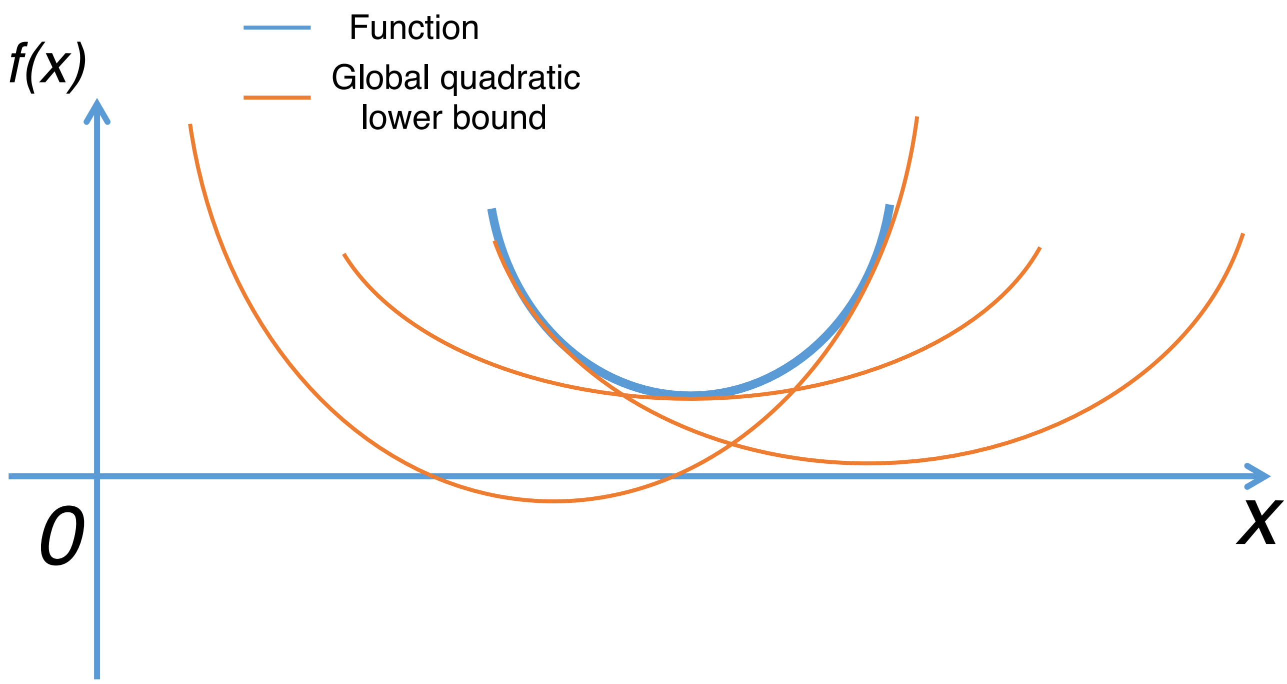
defined on $\{t \mid x + tv \in S\}$ is convex for any $x \in S, v \in \mathbb{R}^n$, which allows to check convexity of the scalar function in order to establish convexity of the vector function.

Strong convexity

$f(x)$, **defined on the convex set** $S \subseteq \mathbb{R}^n$, is called μ-strongly convex (strongly convex) on S, if:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) - \mu\lambda(1-\lambda)\|x_1 - x_2\|^2$$

for any $x_1, x_2 \in S$ and $0 \leq \lambda \leq 1$ for some $\mu > 0$.



Criteria of strong convexity

First order differential criterion of strong convexity

Differentiable $f(x)$ defined on the convex set $S \subseteq \mathbb{R}^n$ is μ-strongly convex if and only if $\forall x, y \in S$:

$$f(y) \geq f(x) + \nabla f^T(x)(y-x) + \frac{\mu}{2}\|y-x\|^2$$

Let $y = x + \Delta x$, then the criterion will become more tractable:

$$f(x + \Delta x) \geq f(x) + \nabla f^T(x)\Delta x + \frac{\mu}{2}\|\Delta x\|^2$$

Second order differential criterion of strong convexity

Twice differentiable function $f(x)$ defined on the convex set $S \subseteq \mathbb{R}^n$ is called μ-strongly convex if and only if $\forall x \in \mathbf{int}(S) \neq \emptyset$:

$$\nabla^2 f(x) \succeq \mu I$$

In other words:

$$\langle y, \nabla^2 f(x) y \rangle \geq \mu \|y\|^2$$

Facts

- $f(x)$ is called (strictly) concave, if the function $-f(x)$ - is (strictly) convex.
- Jensen's inequality for the convex functions:

  $$f\left(\sum_{i=1}^n \alpha_i x_i\right) \leq \sum_{i=1}^n \alpha_i f(x_i)$$

  for $\alpha_i \geq 0; \quad \sum\limits_{i=1}^n \alpha_i = 1$ (probability simplex)

  For the infinite dimension case:

  $$f\left(\int\limits_S x p(x) dx\right) \leq \int\limits_S f(x) p(x) dx$$

  If the integrals exist and $p(x) \geq 0, \quad \int\limits_S p(x) dx = 1$

- If the function $f(x)$ and the set S are convex, then any local minimum $x^* = \arg\min\limits_{x \in S} f(x)$ will be the global one. Strong convexity guarantees the uniqueness of the solution.

Operations that preserve convexity

- Non-negative sum of the convex functions: $\alpha f(x) + \beta g(x), (\alpha \geq 0, \beta \geq 0)$.
- Composition with affine function $f(Ax + b)$ is convex, if $f(x)$ is convex.
- Pointwise maximum (supremum): If $f_1(x), \ldots, f_m(x)$ are convex, then $f(x) = \max\{f_1(x), \ldots, f_m(x)\}$ is convex.
- If $f(x, y)$ is convex on x for any $y \in Y$: $g(x) = \sup\limits_{y \in Y} f(x, y)$ is convex.
- If $f(x)$ is convex on S, then $g(x, t) = t f(x/t)$ - is convex with $x/t \in S, t > 0$.
- Let $f_1 : S_1 \to \mathbb{R}$ and $f_2 : S_2 \to \mathbb{R}$, where $\text{range}(f_1) \subseteq S_2$. If f_1 and f_2 are convex, and f_2 is increasing, then $f_2 \circ f_1$ is convex on S_1.

Other forms of convexity

- Log-convex: $\log f$ is convex; Log convexity implies convexity.
- Log-concavity: $\log f$ concave; **not** closed under addition!
- Exponentially convex: $[f(x_i + x_j)] \succeq 0$, for $x_1, \ldots, x_n$
- Operator convex: $f(\lambda X + (1 - \lambda)Y) \preceq \lambda f(X) + (1 - \lambda) f(Y)$
- Quasiconvex: $f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\}$
- Pseudoconvex: $\langle \nabla f(y), x - y \rangle \geq 0 \longrightarrow f(x) \geq f(y)$
- Discrete convexity: $f : \mathbb{Z}^n \to \mathbb{Z}$; "convexity + matroid theory."

References

- Steven Boyd lectures
- Suvrit Sra lectures
- Martin Jaggi lectures