Optimisation problem $\quad F: \mathbb{R}^n \to \mathbb{R}$ real valued objective function

$$\min F(\vec{x}) \qquad (OPT)$$
$$\text{s.t.} \quad \vec{x} \in \Omega$$

$\Omega \subseteq \mathbb{R}^n$ set of constraints

vector $\vec{x} \in \mathbb{R}^n$

Constraints: ◦) If $\Omega = \mathbb{R}^n$ then (OPT) unconstraind optimisation problem
◦) If $\Omega \subset \mathbb{R}^n$ then inequality and equality constraints

$f_1(\vec{x}) \leq 0, \dots f_m(\vec{x}) \leq 0 \qquad g(\vec{x}) = 0, \dots g_p(\vec{x}) = 0$

m inequality constraints $\qquad$ p equality constraints

Equivalent formulation of (1)

$$\max -F(\vec{x})$$
$$\text{s.t. } \vec{x} \in \Omega$$

A vector $\vec{x}^* \in \mathbb{R}^n$ is called ◦) a global minimiser of (OPT) if $F(\vec{x}^*) \leq F(\vec{x}) \ \forall \vec{x}$

◦) a local minimiser of (OPT) if there exists an open neighborhood $\mathcal{U}$ of $\vec{x}^*$ s.t. $F(\vec{x}^*) \leq F(\vec{x}) \quad \forall \vec{x} \in \mathcal{U} \cap \Omega$ (*)

Local minimum is strict if $F(\vec{x}^*) < F(\vec{x})$

In this module we only consider continuous optimisation, that is F, f_i, g_j $\ i = 1 \dots m, \ j = 1 \dots p$ are continuous.
Other types of optimisation: → discrete optimisation
→ combinatorial optimisation

Examples

i) Linear regression: Given a data set $\{(y_i, x_{i1} \dots x_{ip})\}_{i=1}^N$

y_i: observations; $x_{i1} \dots x_{ip}$: variables (predictors)

Make assumption that relationships is linear, that is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} \qquad \text{for every } i = 1 \dots N$$

Goal: find parameters $\vec{\beta} = (\beta_0, \dots \beta_p)^T$ from data $\ \in \mathbb{R}^{p+1}$

Mean squared loss

$$F(\vec{\beta}) = \frac{1}{N} \sum_{i=1}^{N} \left[y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \right]^2 \qquad (MSL)$$

Rewrite problem in matrix vector form

$$\vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \qquad \vec{X} = \begin{pmatrix} x_{10} & x_{11} & \dots & x_{1p} \\ \vdots & & & \\ x_{N0} & x_{N1} & \dots & x_{Np} \end{pmatrix} \qquad \text{with } x_{i0} = 1 \text{ for } i = 1 \dots N$$

Then (MSL) can be written as

$$F(\vec{\beta}) = \frac{1}{N} \| \vec{y} - \vec{X} \beta \|^2 \qquad \text{where } \|\vec{x}\|_2 = \sum_{i=1}^{d} x_i^2 = \vec{x}^T \vec{x} = \langle \vec{x}, \vec{x} \rangle \quad (MSL)$$

$\|\cdot\|_2$: 2-norm of a vector

(*) Neighborhood of a points contains an open set
see Norms, Metrics & Topologies

$(MSL)_2$ is an unconstrained optimisation problem

$F(\vec\beta)$ ... quadratic function over $\Omega = \mathbb{R}^{p+1}$ $(\vec\beta \in \mathbb{R}^{p+1})$

Write $F(\vec\beta)$ in detail

$$F(\vec\beta) = \frac{1}{N}\|\vec X\vec\beta - \vec y\|^2 = \frac{1}{N}(\vec X\vec\beta - \vec y)^T(\vec X\vec\beta - \vec y)$$

$$= \frac{1}{N}(\vec\beta^T\vec X^T\vec X\vec\beta - 2\vec y^T\vec X\vec\beta + \vec y^T\vec y)$$

$(\vec A\vec B)^T = \vec B^T\vec A^T$

$$= \frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=0}^{p}\sum_{k=0}^{p} x_{ij}x_{ik}\underbrace{\beta_j\beta_k}_{\uparrow\text{ quadratic in }\beta} - 2\sum_{i=1}^{N}\sum_{j=0}^{p} y_i x_{ij}\beta_j + \sum_{i=1}^{N} y_i^2\right]$$

Closed form solution $\vec\beta^* = (\vec X^T\vec X)^{-1}\vec X^T\vec y$ $\Leftarrow$ evaluating / computing this is expensive in practise $(N \gg 1)$ more efficient methods: gradient descent or conjugate gradients

Example: PanTHERIA database contains biological data of 100 740 extant or extinct mammalian species.
Variables include Activity cycle, average lifespan, body mass, gestation length, Litter size, metabolic rate ... (25 in data)
If we pick 573 mammals and look at relationship between only two variables we can observe an approximate relationship

$$y_i = \beta_0 + \beta_1 x_{i1}$$

$$\Rightarrow \left.\begin{array}{ll} \vec y \in \mathbb{R}^N & N = 573 \\ X \in \mathbb{R}^{573\times 2} & p = 1 \end{array}\right\} \Rightarrow \beta_0 = \quad \text{and } \beta_1 =$$

ii) Linear programming: consider airplane with two cargo departments each has weight capacity $WC_1 = 35T$ and $WC_2 = 40t$
$V_1 = 250m^3$ and $V_2 = 400m^3$

	Volume $[m^3/t]$	Weight $[t]$	Profit $[£/t]$
Cargo 1	8	25	£ 300
Cargo 2	10	32	£ 350
Cargo 3	7	28	£ 270

Variables x_{ij} how much of cargo i goes into compartment j (in tonnes)

$\vec x = (x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32})^T$

Total profit

$$F(\vec x) = 300(x_{11} + x_{12}) + 350(x_{21} + x_{22}) + 270(x_{31} + x_{32})$$

Constraints:

$x_{11} + x_{12} \leq 23$ (total amount of cargo 1)
$x_{21} + x_{22} \leq 32$ (— " — 2)
$x_{31} + x_{32} \leq 28$ (— " — 3)
$x_{11} + x_{21} + x_{31} \leq 35$ (weight constraint of compartment 1)
$x_{12} + x_{22} + x_{32} \leq 40$ (— " — 2)

$8x_{11} + 10x_{21} + 7x_{31} \le 250$ (volume constraint comp 1)

$8x_{12} + 10x_{22} + 7x_{32} \le 400$ (— " — 2)

$\frac{x_{11}+x_{21}+x_{31}}{35} - \frac{x_{12}+x_{22}+x_{32}}{40} = 0$ Balance in weight ratio

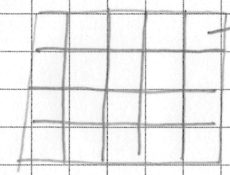
$x_{ij} \ge 0$ negative weights make no sense

Write problem again in matrix vector form

$$\begin{aligned} \min\ & \langle \vec c, \vec x\rangle \\ \text{s.t. } & \vec A \vec x \le \vec b \\ & \vec B \vec x = \vec d \\ & \vec x \ge 0 \end{aligned}$$

Component-wise

iii) Image inpainting



Pixel either corresponding to Light intensity $[0,255]$ (greyscale) or a color vector (usually RGB)

Image $u \in \mathbb{R}^{m\times n}$ is corrupted or some information is missing

Goal: find an image that is close to the original image $\vec u$

Total variation norm

$$\|\vec u\|_{TV} = \sum_{i=1}^{N}\sum_{j=1}^{N} \sqrt{\underbrace{(u_{i+1,j}-u_{ij})^2}_{\text{vertical differences}} + \underbrace{(u_{i,j+1}-u_{ij})^2}_{\text{horizontal differences}}}$$

$u = (u_{ij})_{i=1..n,\ j=1..m}$

Consider $\vec u_1 = \begin{pmatrix} 0 & 17 & 3 \\ 7 & 32 & 0 \\ 2 & 9 & 27 \end{pmatrix}$ $\vec u_2 = \begin{pmatrix} 1 & 1 & 3 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}$

$\|\vec u_1\|_{TV} = 200.637 > \|\vec u_2\|_{TV} = 14.721$ usually

Images with artefacts & noise have higher TV norm

Let $u \in \mathbb{R}^{m\times n}$ be an image

$\Omega \subset [m]\times[n] = \{(i,j): 1 \le i \le n,\ 1 \le j \le m\}$ set of indizes where corrupted and original image coincide

convex function

$$\begin{aligned} \min\ & \|\vec x\|_{TV} \qquad \text{(TV)} \\ \text{s.t. } & x_{ij} = u_{ij} \text{ for } (i,j) \in \Omega \end{aligned}$$

⇐ don't change pixels that have right value

(TV) is an example of a convex optimisation problem