EngSci 721

Inverse Problems and Learning From Data

Oliver Maclaren (oliver.maclaren@auckland.ac.nz)

1. *Basic concepts* [5 lectures + 1 Tutorial]
   Forward vs inverse problems. Well-posed vs ill-posed problems. Algebra and calculus of inverse problems (left and right inverses, generalised and pseudo inverses, resolution operators, matrix calculus). Representing higher dimensional problems (image data etc).
2. *Instability and regularisation in linear and nonlinear problems* [6 lectures + 1 Tutorial]
   Instability and related issues for generalised inverses. Introduction to regularisation and trade-offs. Tikhonov regularisation. Higher-order Tikhonov regularisation. Sparsity and regularisation using different norms. Truncated singular value decomposition. Iterative regularisation, including stochastic/mini-batch gradient descent.
3. *Further topics* [3 lectures + 1 Tutorial]
   Regularisation parameter choice, including statistical and machine learning views of regularisation. Confidence sets for linear and nonlinear models. Physics-informed machine learning and neural networks.

Module overview

Inverse Problems and Learning From Data (*Oliver Maclaren*)
[~14 lectures/3 tutorials]

Lecture 2: Inverses I

Topics:

- Algebra of inverse problems
- Tall and wide systems
- Least squares solutions to tall systems
- Least norm solutions to wide systems
- Left and right inverses

EngSci 721 : Lecture 2. Inverses I

- Algebra of inverse problems (IPs)
  (cf eg calculus of IPs)

  ↳ Resolving lack of existence & /or uniqueness
    ↳ non-square linear systems
    ↳ non 1-1/onto nonlinear systems

  ↳ Left & right inverses

  ↳ Formulating as optimisation problems
    ↳ least squares & least norm solutions (today)
    ↳ matrix calculus ? (later)

  ↳ Algebraic characterisation of various types of inverse: preview

Algebra of Inverse Problems

Generalised Inverses

Our basic problem can be defined as:

> 'solve', ie 'invert', equations like $F(x) = y$ for x, given y

where:

- x & y could be vectors, functions, images etc
- solutions might not exist, might not be unique &/or might not be stable

Note: mappings, measurements & vectors

We might have an 'exact' model of the form

$$\boxed{y = a + bx}, \text{ for } \boxed{\text{scalars}}\ x, y$$

If we take a series of 'noisy' $\boxed{\text{measurements}}$ we get eg:

$$y_{obs,i} = (a+bx)_i + e_i$$

'noise' for obs i

$$\Rightarrow \quad y_{obs,i} = a + x_i + e_i$$

for $i = 1, \ldots, n$.

These lead to $\boxed{\text{vector}}$ eqns in terms of the noisy observations/realisations:

$$\begin{bmatrix} y_{obs,1} \\ \vdots \\ y_{obs,n} \end{bmatrix} = a \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + b \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}}_{\text{don't know true errors!}}$$

$$\Rightarrow \boxed{\bar{y}_{obs} \approx a\bar{1} + b\bar{x}} \Big\} \ \bar{y}, \bar{x}, \bar{1} \text{ vectors}$$

solve approx. since e unknown

(I usually drop the explicit overbars on vectors)

Linear or nonlinear? Finite or Infinite?

- We will discuss some basic algebra of the problem in the linear & finite dimensional setting

  → ie using Linear Algebra in $\mathbb{R}^n$

- This can be considered as the 'algebra of linear mappings' in $\mathbb{R}^n$

→ An important question is:

do these results carry over to the nonlinear &/or infinite-dimensional setting(s)?

Short answer: yes!

(key concepts & as long as careful)

Linear or nonlinear? Finite or Infinite?

- Algebra of arbitrary mappings?

  ↳ Category theory

  (see eg Nashed / Maclaren & Nicholson for generalised inverses in category theory setting)

- Discontinuous infinite dimensional linear mappings ('ill-posed')

  ↳ Ill-conditioned finite dimensional matrices

→ Theory is a bit beyond scope, but most of the key tools are applicable to nonlinear setting (we'll solve some of these!)

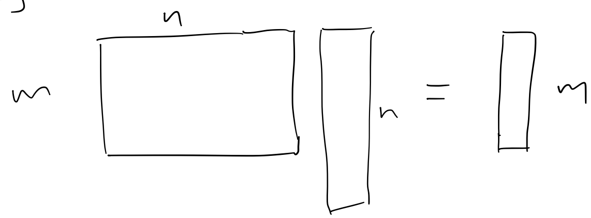
Linear setting

$$\mathbb{R}^n \underset{?}{\overset{A}{\rightleftarrows}} \mathbb{R}^m$$

Consider the system of equations

$$Ax = y \quad \left\}\begin{array}{l} - A \text{ is } m\times n \text{ matrix} \\ - x \in \mathbb{R}^n \text{ vector (inputs)} \\ - y \in \mathbb{R}^m \text{ vector (outputs)} \end{array}\right.$$

eg



rows : eq^{n}s

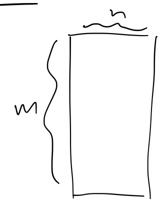
cols : unknowns

→ can we solve? '$Ax = y$'?

- How do we solve when $m \neq n$?

  → $m > n$, more rows than cols. } $\boxed{\text{tall}}$

existence?



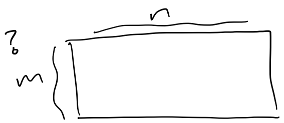
→ eq^{n}s > unknowns

→ possibly inconsistent / overdeterm.

→ 'more data than parameters'

  → $m < n$, more cols than rows } $\boxed{\text{wide}}$

uniqueness?



→ unknowns > eq^{n}s

→ possibly many sol^{n}s

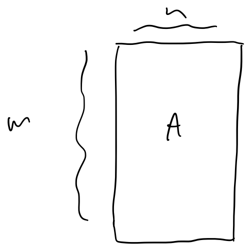
→ 'more param. than data'

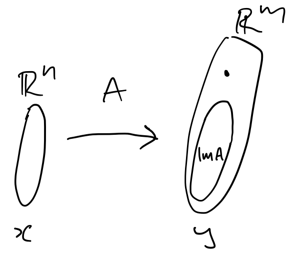
Tools developed to 'solve' each case

Case 1: possibly inconsistent: more equations (rows) than unknowns (cols)

↳ $\boxed{\text{existence}}$ issue

Consider '$Ax = y$', A is $m \times n$ & $m > n$ } $\boxed{\text{'Tall'}}$





→ Assume for now that all n columns are linearly independent (see handout)

↳ 'tall' with LI col. linear version of 1-1 function

- Then, if also have $\boxed{y \in \operatorname{Im} A}$ there is a $\boxed{\text{unique}}$ solution

Im A: range/col space/image of A, ie $A[\mathbb{R}^n] = \{y \in \mathbb{R}^m \mid \exists x \in \mathbb{R}^n, Ax = y\}$

- Instead, if $\boxed{y \notin \operatorname{Im} A}$

⇒ no (exact) solution: doesn't $\boxed{\text{exist}}$

⟶

If $y \notin \operatorname{Im} A$, no solⁿ: 'Inconsistent equations'

⇒ Change goal to approximate solution of '$Ax = y$'

↳ resolves lack of existence

→ Define $r = y - Ax$ } residual 'error' (obs. − model)

→ Measure size of error with a norm $\|\cdot\|$ (see handout for diff. types)

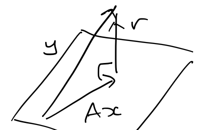
→ Typically assume $\|\cdot\|_2$

New problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimise}} \; \|y - Ax\|, \quad A \,\&\, y \text{ given}$$

- "best approximation"
- "closest approx"

etc.



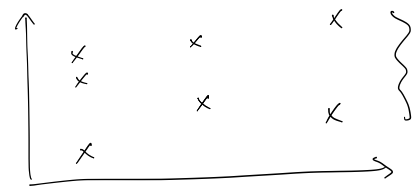
} 'project onto range of A'

Minimiser of $\|\ \|_2$ & minimiser of $\|\ \|_2^2$ are same ($x \mapsto x^2$ is monotonic for $x \geq 0$)

⇒ Least squares approximation:

$$\underset{x}{\text{minimise}} \; \|y - Ax\|_2^2 \quad \text{(equiv. problem)}$$

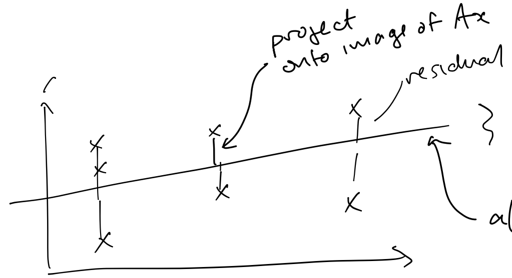
Illustration



no $\overset{\sim Ax}{\text{line}}$ can fit all

→ inconsistent

→ eg due to random error, $e = y - Ax$



least squares

all outputs of Ax look like lines

Derivation? We will learn how later!

$$\min_x \|y - Ax\|_2^2 = \min_x \langle y - Ax, y - Ax \rangle$$

$$= \min_x f(x)$$

where

$$f(x) = \langle y - Ax, y - Ax \rangle$$
$$= (y - Ax)^T (y - Ax)$$

same, different notation.

$$= \langle y, y \rangle - 2 \langle y, Ax \rangle + \langle Ax, Ax \rangle$$

$$= y^T y - 2 y^T A x + x^T A^T A x$$

$$= y_i y_i - 2 y_i A_{ij} x_j + (A_{ij} x_j)(A_{ik} x_k)$$

etc

Note: Norms, products, summation etc

- $$\|x\|_2^2 = \langle x, x \rangle = x^T x = \sum_i x_i^2$$

- $$Ax = \sum A_{ij} x_j = A_{ij} x_j$$ (Einstein summation convention → sum over repeated indices)

So:

- $$y^T A x = y_i A_{ij} x_j$$

See handout (we'll do some practice too!)

To solve analytically, need to be able to...

differentiate vectors, matrices, tensors wrt vectors, matrices, tensors ...

Requires:

Matrix calculus', Tensor calculus etc

→ multiple conventions / notation

(see eg Wiki page on Matrix Calculus)

→ we will look at this & provide eg 'three key rules' .... in later lecture

For now, let's focus on key end results & big picture

least squares: skipping derivation detail for now..

$$\min_x f(x) = y^T y - 2y^T A x + x^T A^T A x$$

$\Rightarrow$ set $D_x f = 0^T$ (1) (row vector)

(or $\nabla_x f = 0$ ...)

⋮

matrix calculus

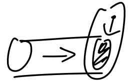
⋮

$$(1) \Rightarrow -2y^T A + 2x^T A^T A = 0^T$$

$$\Rightarrow -A^T y + A^T A x = 0$$

$$\Rightarrow \boxed{A^T A x = A^T y}$$

Least squares approximation



'data reduction'

This hence leads to....

The normal equations:

$$\boxed{A^TAx = A^Ty}$$

normal? $A^T(Ax-y)=0$
geometric orthog of error

→ Since we assume the n cols of A are linearly independent then A^TA is invertible (see handout) & so get unique approximate sol^n

$$\rightarrow \boxed{x^* = (A^TA)^{-1}A^Ty} \leftarrow$$

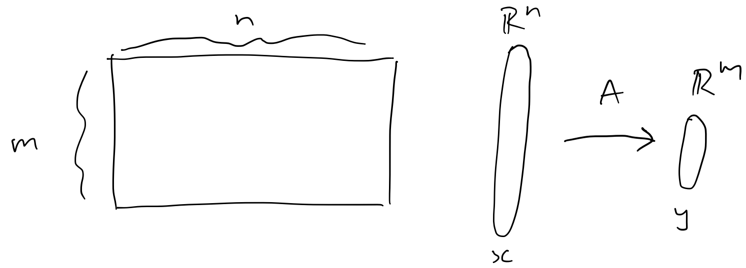
(we will look at what happens if not LI soon!).

Now consider 'dual' prob....

Case 2: possibly non-unique: more unknowns than equations

↳ uniqueness issue

Consider $Ax = y$, A is $m \times n$ & $m < n$ } 'wide'



→ Assume for now that all m rows are linearly independent

↳ wide with LI rows: linear version of onto function

→ Assume $Ax = y$ has at least one solution, ie $y \in \operatorname{Im} A$ (range / image of A)

General solution of $Ax = y$:

$$\boxed{x = x^* + x_0}$$

where $x_0 \in \ker A$ & x^* is any sol^n of $Ax = y$

(Recall: 'kernel', $\ker A$ = null space of A = $\{x \in \mathbb{R}^n \mid Ax = 0\}$)

→ Underdetermined equations : least norm sol^{n}s

- If we want to 'pick out' a single sol^n (we don't always!), a 'natural' choice is to choose the 'smallest' (think : 'simplest' or 'most efficient' sol^n) in some sense

- Again, this is relative to a particular norm, eg $\|\cdot\|_2 = l_2$ norm, & other factors like regularisation details.

⇒ Problem to solve for 'least norm' sol^n:

$$\min_x \|x\|$$
$$\text{st. } Ax = y$$

note: here assuming exactly solvable cf before

Equivalent to:

$$\left.\begin{array}{l}\min_x \|x\|^2 \\ \text{st. } Ax = y\end{array}\right\}$$

'least squares sol^n' (cf least squares approximation)

Least norm problem ('Model reduction')

Using duality or lagrange multipliers (see later?) you can show that the least-squares solution to the minimum norm problem requires solving the dual problem

$$\boxed{AA^T v = -2y \quad \text{for } v^*}$$

& then obtaining $\boxed{x^* = -\frac{1}{2} A^T v^*}$

derived later

→ Since we assume the m rows of A are linearly independent then AA^T is invertible (see handout) & so get unique minimum norm

$$\rightarrow \boxed{x^* = A^T (AA^T)^{-1} y} \leftarrow$$

(we will look at what happens if not LI soon!).

Summary so far:

- $\boxed{\text{Tall}}$ / overdetermined (+ LI cols): can find least squares $\boxed{\text{approx}}$ } resolve lack of existence
- $\boxed{\text{Wide}}$ (underdet.) (+ LI rows): can find least squares/norm $\boxed{\text{sol}^n}$ } resolve lack of uniqueness

---

→ no proper 'full' inverse exists in each case, but each is either an algebraic $\boxed{\text{left inverse}}$ or $\boxed{\text{right inverse}}$

$$\boxed{\begin{array}{ll} LA = I & \text{(left)} \\ AR = I & \text{(right)} \end{array}}$$

→ These solve rectangular systems from 'one direction/side'.

→ cont'd tomorrow!

Example

Projectile motion (Aster et al Ex. 1.1)

$$\boxed{y(t) = a + bt - 0.5ct^2}$$

→ estimate a, b, c given we observe at m times: } def $\theta = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$

$\theta \in \mathbb{R}^n = \mathbb{R}^3$

$$\begin{aligned} y_1 &= a + bt_1 - 0.5t_1^2 \\ y_2 &= a + bt_2 - 0.5t_2^2 \\ &\vdots \\ y_m &= a + bt_m - 0.5t_m^2 \end{aligned}$$

ie
$$\begin{bmatrix} 1 & t_1 & -0.5t_1^2 \\ 1 & t_2 & -0.5t_2^2 \\ \vdots & & \\ 1 & t_m & -0.5t_2^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

ie $\boxed{F \cdot \theta = y}$

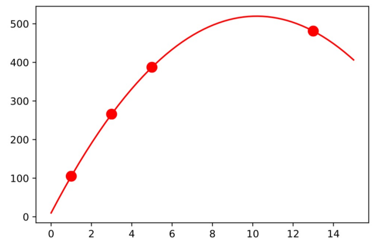
[Note: linear in parameters!]
& using θ instead of x

Example: Least squares (param < data)

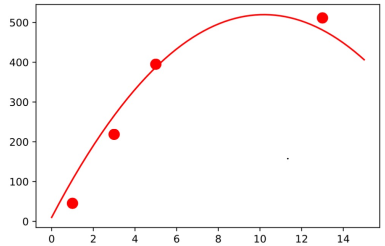
→ more than three observations y
eg 4 obs. (y), 3 param (θ)

→ noise means doesn't satisfy $A\theta$ exactly
↳ eg $y = A\theta + e$

```python
def fmap(tobs):
    A = np.zeros((len(tobs),3))
    for i, ti in enumerate(tobs):
        A[i,:] = np.array([1,ti,-0.5*ti**2])
    return A

#true parameters
theta_true = np.array([10,100,9.81])

#fine time grid
t = np.linspace(0,15,1000)

#observation times
#tobs = np.array([1,13]) #under-determined
tobs = np.array([1,3,5,13]) #over-determined

#forward map
Aobs = fmap(tobs)

#observed data
yobs = np.dot(Aobs,theta_true) #noise-free
#yobs = np.dot(Aobs,theta_true) + np.random.normal(0,30,size=len(tobs))

#plots
plt.plot(tobs,yobs,'ro',markersize=10)
plt.plot(t,x,'r')
plt.show()
```



noise free
$A\theta$



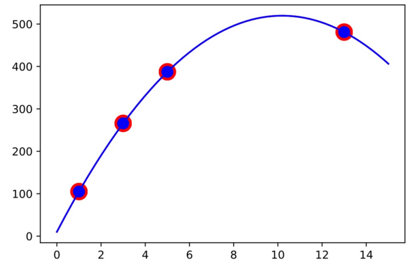
noisy.
$A\theta + e$

Invert:

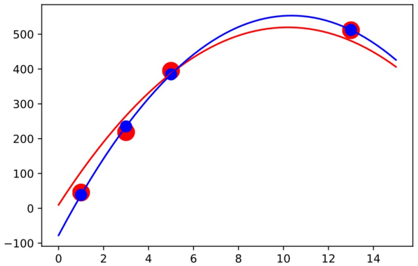
pseudo inverse, here left inverse

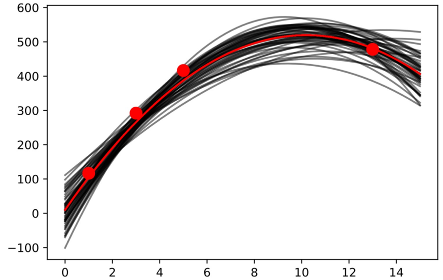
```python
Apinv = np.linalg.pinv(Aobs)
```

noise free



noisy





repeated sampling

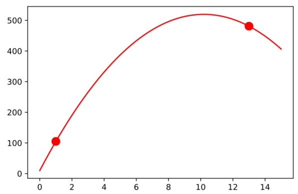
Exercise: explore what the parameters are doing compared to true values (this & next)

Example : least norm (param > data)
$\theta \qquad y$

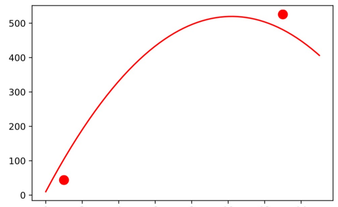
→ less than three observations

Solutions

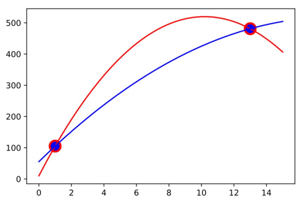
noise free data



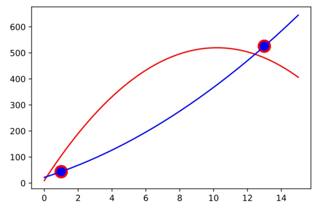
noisy data



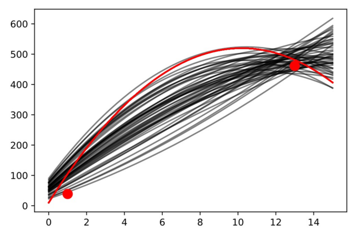
recovered (blue)



recovered (blue)



- data fit perfectly
- sol^n bad



repeated sampling

What about parameters?

Note: we've used the naive model norm $\|\theta\|_2$ including the constant term → here curved actually 'smaller' than straight (see later)

Exercises

- Consider the approximate polynomial model :

$$y = a + bx + cx^2 + dx^3$$

& data $(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)$

- show how to formulate the problem of determining the parameters a, b, c, d as an approximate matrix eq^n
- if $m > 4$ is this a tall or wide system?
- In what sense is the above case 'solvable'?