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 3: Inverses II

Topics:

- From left and right inverses to generalised and pseudo-inverses
- Projection and resolution operators

EngSci 721 : Lecture 3 Inverses II

- From left & right inverses to generalised (& pseudo) inverses
- Projections & resolution operators

Recall:

Algebra of Inverse Problems

Our basic problem can be defined as:

$$\boxed{\text{'solve', ie 'invert', equations like } F(x) = y \text{ for } x \text{, given } y}$$

where:

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

Recall

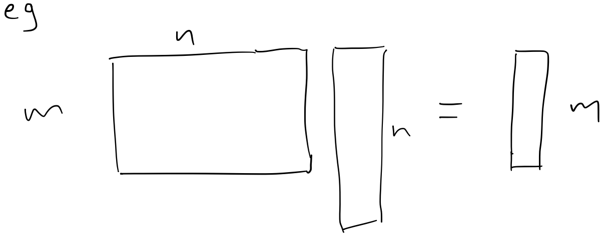
Linear Setting

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

Consider the system of equations

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

eg

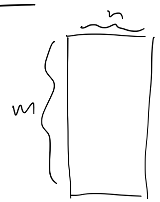


rows : eq^{n}s
cols : unknowns

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

→ $m > n$, more rows than cols.

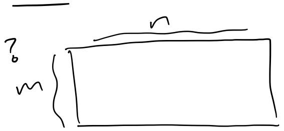
existence?



→ eq^{n}s > unknowns
→ possibly inconsistent/overdeterm.
→ 'more data than parameters'

→ $m < n$, more cols than rows

uniqueness?



→ unknowns > eq^{n}s
→ possibly many sol^{n}s
→ 'more param. than data'

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 $\boxed{\text{right inverse}}$

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

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

Left inverses : Algebra

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

L is a left inverse (a retraction in nonlinear/category theory) for A if it satisfies

$$\boxed{LA = I} \quad \left\{ \begin{array}{l} A : m\times n \\ L : n\times m \\ I : n \times n \ \} \text{ also write } I_n \end{array} \right.$$

For least squares data approximation, we have, when columns of A are linearly independent:

$$\boxed{L = (A^TA)^{-1}A^T} \ \& \ \boxed{x = Ly}$$

Notes:

- the linear indep. condition ensures A^TA is invertible, ie $(A^TA)^{-1}$ exists!
- a left inverse for a tall system exists if & only if the cols of A are linearly independent
- $LA = (A^TA)^{-1}A^TA = G^{-1}G = I$ ✓ for $\underbrace{G = A^TA}_{\text{the 'Gram matrix'}}$

Suppose a left inverse exists, so

$A : m\times n$
$L : n\times m$

$$LA = I : \quad \begin{array}{cc} & m \\ n & \boxed{L} \end{array} \ \begin{array}{cc} & n \\ m & \boxed{A} \end{array} = \begin{array}{cc} & n \\ n & \boxed{I} \end{array}$$

What does AL equal? Dimensions ok:

$$\begin{array}{cc} & n \\ m & \boxed{A} \end{array} \ \begin{array}{cc} & m \\ n & \boxed{L} \end{array} = \begin{array}{cc} & m \\ m & \boxed{?} \end{array}$$

→ In general $AL \neq I$

But consider $(AL)^2 = (AL)(AL)$

$$(AL)(AL) = A\underbrace{(LA)}_{I}L = AIL = AL$$

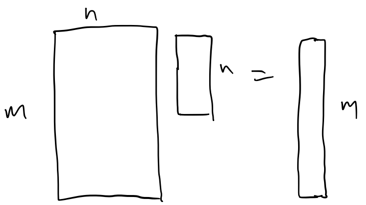
So $\boxed{P = AL}$ satisfies $\boxed{P^2 = P}$ ('idempotent')

→ this is the defining property of a projection matrix!

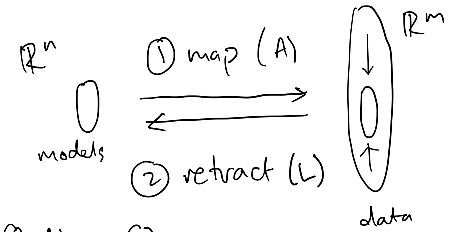
→ Note $I^2 = I$ as well, tho $P \neq I$ here

So:

A left inverse exists when # rows $\geq$ # cols of A $\boxed{\&}$ the cols are LI (think 1-1 map into bigger space)



Effects of LA & AL look like:



'shrink data to make solvable'

① then ② is identity on model space $\boxed{LA = I}$

vs.

② then ① shrinks/projects $\mathbb{R}^m$ data to subspace of $\mathbb{R}^m$ $\boxed{AL = \text{projection}}$.

Note: AL maps from $\mathbb{R}^m$ (data) to $\mathbb{R}^m$ (data)

→ projects onto image of A

Right inverses: Algebra

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

R is a right inverse (a section in nonlinear/category theory) for A if it satisfies

$$\boxed{AR = I} \quad \left\{ \begin{array}{l} A: m \times n \\ R: n \times m \\ I: m \times m \end{array} \right.$$

For least norm problems, we have, when rows of A are linearly independent:

$$\boxed{R = A^T(AA^T)^{-1}} \quad \& \quad \boxed{x^* = Ry}$$

— Notes:

→ linear independence of rows of A
  $\iff$ linear independence of cols of A^T

→ ensures $(AA^T)^{-1}$ exists
  ↳ AA^T is Gram matrix of A^T!

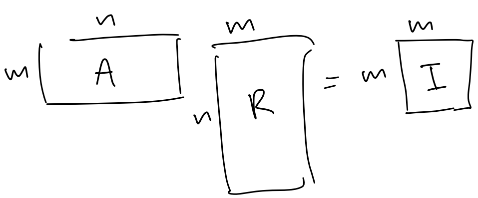
→ a right inverse for a wide system exists if & only if the rows of A are linearly independent (or cols of A^T)

→ $AR = A(A^T(AA^T)^{-1}) = (AA^T)(AA^T)^{-1} = I$ ✓

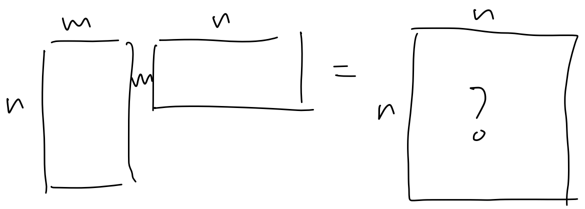
Suppose a right inverse exists, so

$A : m \times n$
$R : n \times m$
$I : m \times m$

$AR = I$:



What does RA equal? Dimensions ok:



→ In general $RA \neq I$

But consider $(RA)^2 = (RA)(RA)$

$(RA)(RA) = R(AR)A = RIA = RA$

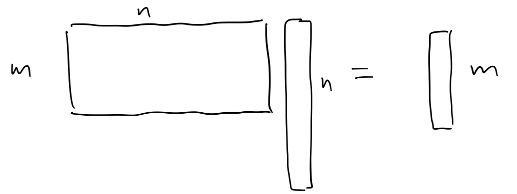
So $\boxed{P = RA}$ satisfies $\boxed{P^2 = P}$ ('idempotent')

→ this is the defining property of a projection matrix! (again!)

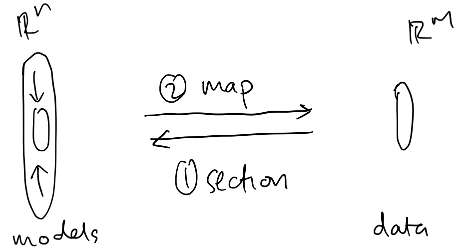
→ Note $I^2 = I$ as well, tho $P \neq I$ here

So:

A right inverse exists when # cols ≥ # rows of A & the rows are LI (think onto mapping from big to small space)



Effects of AR & RA look like:



'shrink models to make unique'

② then ① while ① then ② is identity on data space

'shrinks'/projects models to subspace

$\boxed{RA = \text{projection}}$

$\boxed{AR = I}$

Note: RA maps from $\mathbb{R}^n$ (model) to $\mathbb{R}^n$ (model)

→ projects onto 'model' (parameter/input) space.

Unification & extension: Generalised (& pseudo) inverses

→ so far we required linearly indep. cols/rows for our tall/wide systems in order to get left/right inverses

→ we can extend to general case + unify the solution of these problems with 'generalised/pseudo' inverses'

→ This will solve both existence & uniquess issues at the same time!
  - spoiler alert: but not stability issues!

- do not require lin. indep. cols/rows
  - always exist regardless
- not left or right inverse in general tho
  - next closest thing
  - projection on both spaces in gen.

Generalised inverses: optimisation formulation

- Generalised (or pseudo, for the special cases we look at) inverses solve the two step optimisation prob:

- Stage 1: $\underset{x}{\text{minimise}} \; \|y - Ax\|$ or $\|y - Ax\|^2$

Then

- Stage 2: minimise $\|x\|$ or $\|x\|^2$ among all solutions to stage 1.

The result is a matrix A^{+} ← dagger symbol or plus that

→ is a left inverse if one exists
→ is a right inverse if one exists
→ 'close to' left/right inverse if neither exist.
  - eg cols/rows not LI

Note: Optimisation form easier to generalise to nonlinear

Generalised inverses

The most general algebraic characterisation of a generalised inverse A^+ is just

(1) $\boxed{AA^+A = A}$

- This is not unique tho' & we can add additional conditions

→ the most general additional condition is to require

(2) $\boxed{A^+AA^+ = A^+}$ (A is gen. inverse to A^+!)

Then:

- $\underbrace{(AA^+)(AA^+)}_{A \text{ via (1)}} = AA^+$ ie AA^+ is $\boxed{\text{projection}}$
- $\underbrace{(A^+A)(A^+A)}_{A^+ \text{ via (2)}} = A^+A$ ie A^+A is $\boxed{\text{projection}}$

→ gives ...

Model resolution, data resolution operators:

$\boxed{R_D = AA^+}$ } how much data is 'shrunk' or smeared } 'data resolution' operator

$\boxed{R_M = A^+A}$ } how much model is 'shrunk' or smeared } 'model resolution' operator

---

Recall: Projection operators P characterised by

$\boxed{P^2 = P}$ ('idempotent')

→ one application of P gives 'maximum' effect

→ Not I in general, but something 'similar'

→ Note $I^2 = I$ ('idempotent')

→ Want to compare to identity I (full resolution)

As above, R_D is a projection on data space

R_M is a projection on model/param. space

$\boxed{R - I \text{ measures 'bias'}}$ } · look at diagonals · resol. test.

for $R = \begin{cases} R_D \\ R_M \end{cases}$

Generalised vs pseudo-inverses

- Generalised inverses defined via (1) & (2) are still not unique

→ add extra conditions } eg different norms etc
→ depends on goals } diff. inverses

Pseudo?

Eg what we've been using (least sq/norm) →(& will assume unless otherwise) is the Moore–Penrose pseudo-inverse

This satisfies

gen. inv.: $AA^{+}A = A$, $A^{+}AA^{+} = A^{+}$

$A^{+}A$ symm.: $(A^{+}A)^{T} = A^{+}A$

AA^{+} symm.: $(AA^{+})^{T} = AA^{+}$

least squares, least norm generalised inverse = 'pseudo-inverse'

→ pseudo-inverse gives a symmetric projection on each space, where

symmetric proj. = orthogonal proj

Least squares / least norm?

Our optimisation def^n actually gives the pseudo inverse, ie solving:

- Stage 1: minimise$_{x}$ $\|y - Ax\|$ or $\|y - Ax\|^2$

Then

- Stage 2: minimise $\|x\|$ or $\|x\|^2$ among all solutions to stage 1.

Implicitly defines an A^{+} that satisfies the Moore-Penrose pseudo inverse conditions

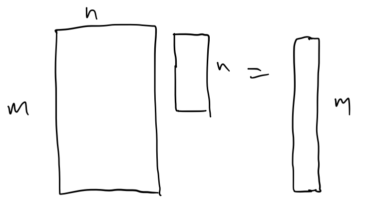
→ When they exist, A^{+} pseudo inverse gives left/right inverses: $L = (A^{T}A)^{-1}A^{T}$, $R = A^{T}(AA^{T})^{-1}$

→ AL & RA are orthogonal projections as $(A(A^{T}A)^{-1}A^{T})^{T} = A(A^{T}A)^{-1}A^{T}$ etc.

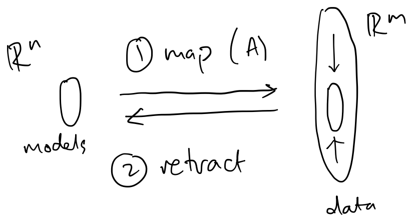
Summary:

Least squares data approx

(A^+ is left inverse in this case, ie linearly indep. col.)



ie



$R_D \neq I$

$R_M = I$

can recover noise free models exactly

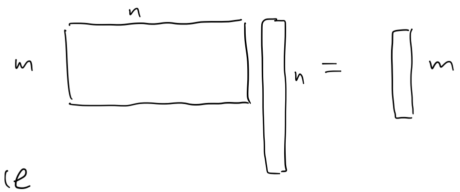
ie 1-1 map.

→

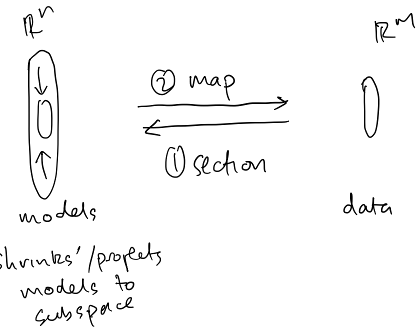
Summary:

Least squares model reduction

(A^+ is right inverse in this case, ie linearly indep. rows)



ie



$R_D = I$

$R_M \neq I$

can fit data exactly, ie onto map

Subtle point (see later):

we often actually want $R_D \neq I$ in above, if we have noise!

↳ ie A^+ solves existence/uniqueness but not stability!

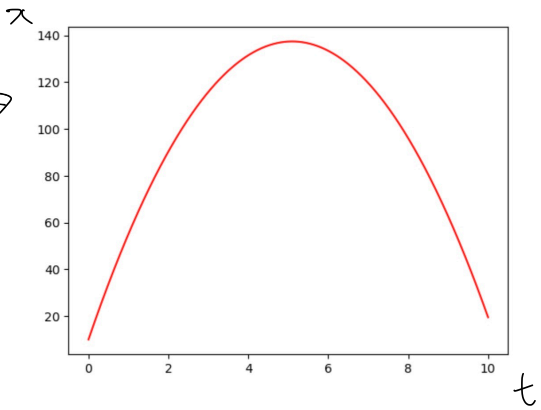
Example

consider projectile problem again and look at R_M, R_D vs I: now close to 'full resolution'

→ plot/evaluate matrices

→ evaluate for given models/data

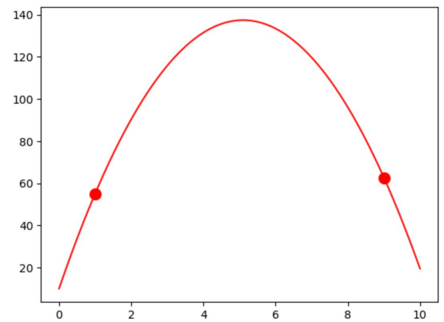
```
import numpy as np
import matplotlib.pyplot as plt

# Projectile motion
t = np.linspace(0,10,1000)
t_max = np.max(t)
def model(t,a,b,c):
    #note: scale initial velocity by 0.5*t_max to put on similar scale
    return a + (0.5*t_max)*b*t - 0.5*c*t**2
vmodel = np.vectorize(model)

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

#input 'signal'
x = vmodel(t,*x_true)

#plot
plt.plot(t,x,'r')
plt.show()
```



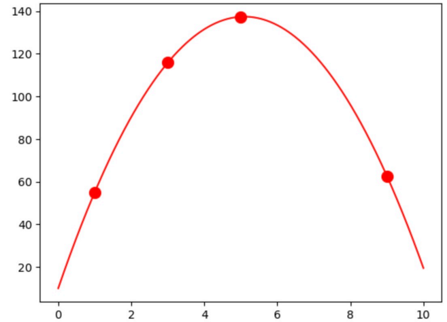

```
#matrix form
def fmap(tobs):
    A = np.zeros((len(tobs),3))
    for i, ti in enumerate(tobs):
        A[i,:] = np.array([1,0.5*t_max*ti,-0.5*ti**2])
    return A

#fine time grid
t = np.linspace(0,10,1000)

#observation times
tobs = np.array([1,9]) #under-determined
#tobs = np.array([1,3,5,9]) #over-determined

#forward map
Aobs = fmap(tobs)

#observed, noise-free data
yobs = np.dot(Aobs,x_true) #noise-free

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

under & over

under det.



over det.



```python
# pseudo-inverse
Apinv = np.linalg.pinv(Aobs)

#Model resolution
R_model = Apinv @ Aobs
print(R_model)
plt.imshow(R_model,extent=[0,3,3,0],cmap='Blues')
plt.axis('off')
plt.colorbar()
plt.clim(0,1)

#data resolution
R_data = Aobs @ Apinv
print(R_data)
plt.imshow(R_data,cmap='Blues')
plt.colorbar()
plt.axis('off')
plt.clim(0,1)
```

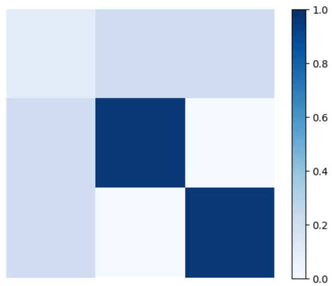
underdet.:

Rm: x_0 v_0 g

x_0

v_0

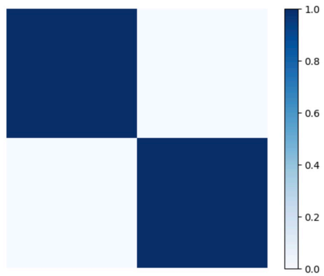
g



Rd: t= 1 9

t=1

9



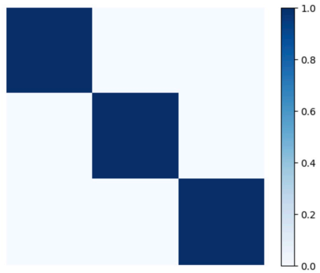
over det.

Rm: x_0 v_0 g

x_0

v_0

g



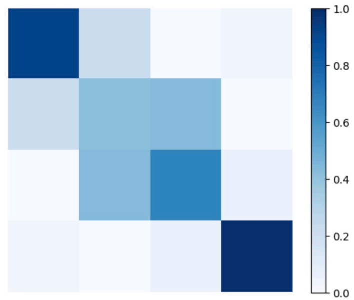
Rd: 1 3 5 9

t=1

3

5

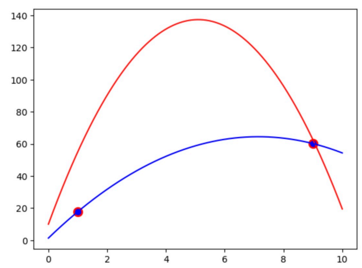
9




```python
#model resolution test for given model
print('true model')
print(x_true)
print('resolved model')
print(R_model@x_true)
print('relative bias (%)')
print(100*np.abs(R_model@x_true - x_true)/x_true)

#data resolution test for noise-free data
#note: R_data@yobs = A_obs@Apinv@yobs = A_obs@x_est

#projected data points and full noise-free model
plt.plot(tobs,yobs,'ro',markersize=10)
plt.plot(t,x,'r')
plt.plot(tobs,R_data@(yobs),'bo')
plt.show()

#projected data points and full noisy-model
y_noisy = yobs + np.random.normal(0,30,size=len(tobs))
plt.plot(tobs,y_noisy,'ro',markersize=10)
plt.plot(tobs,R_data@(y_noisy),'bo')
plt.plot(t,x,'r')
#include implied predictions for other time points:
# evaluate model on fine grid given estimated parameters
Apred = fmap(t)
plt.plot(t,Apred@Apinv@y_noisy,'b')
plt.show()
```

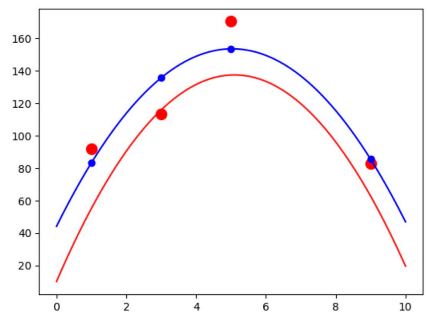
underdet:

```
true model
[10.   10.    9.81]
resolved model
[ 4.90539326 11.13213483 10.94213483]
relative bias (%)
[50.94606742 11.32134831 11.5406201 ]
```



overdet.

```
true model
[10.   10.    9.81]
resolved model
[10.   10.    9.81]
relative bias (%)
[6.75015599e-13 7.10542736e-14 9.05380652e-14]
```



Exercises

Consider the linear inverse problem of 'solving' $Ax = y$ for x given y & A

- Suppose either a right or left inverse exists as appropriate
  - State & prove a property these inverses share with the identity (when they exist)
  - Describe the effects of RA & AL & the spaces they map between (when they exist)
  - Give expressions for the data & model (parameter) resolution matrices in terms of A & A^+
  - How do these relate to the previous question?