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 12: Regularisation parameter choice

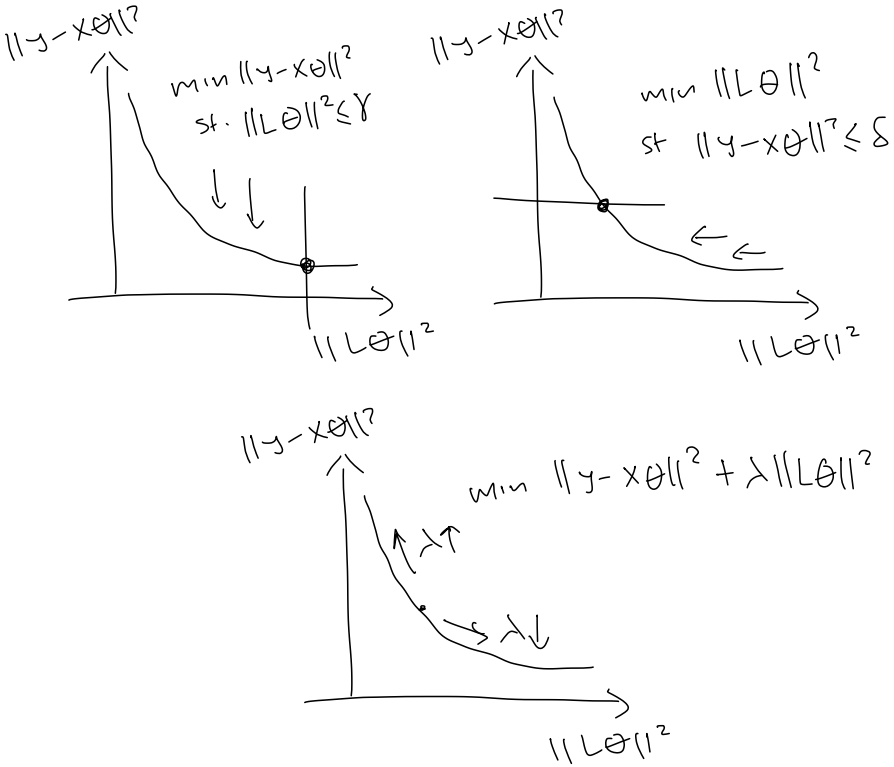
Topics:

- Discrepancy principles
- Cross-validation
- Further reading

Eng Sci 721 : Lecture 12.

- Regularisation parameter choice methods
  - Discrepancy approaches
  - Cross-validation
  - Further reading

Recall three ways to think about trade-off curves:



These are 'equivalent' in the sense that they give the same sol^n for appropriate choice of λ, δ, γ

Discrepancy principles

The general idea of a discrepancy principle is to solve eg

$$\begin{cases} \min \|L\theta\|^2 & \} \text{ model norm term} \\ \text{st. } \|y - X\theta\|^2 \leq \delta & \} \text{ data norm term} \end{cases}$$

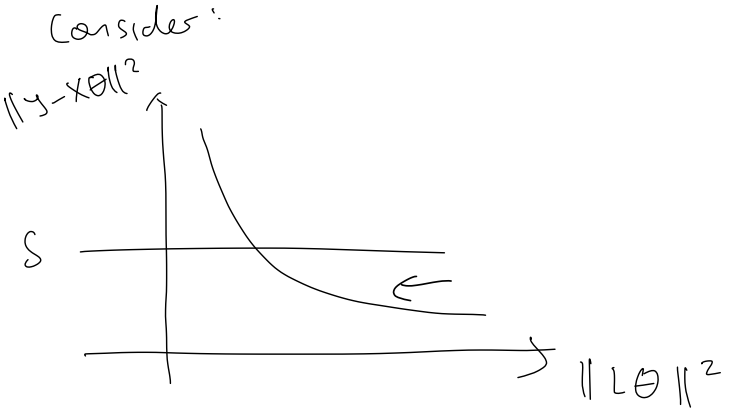
for some choice of δ

Assuming the constraint is binding this means we enforce $\|y - X\theta\|^2 = \delta$ for some δ.

Different methods of choosing δ exist. Here is one that has a nice interpretation (imo!)

↳ 'simplest adequate/acceptable fitting model'

Consider:



In some cases we may have info on δ, eg if we have a statistical model $Y_i = X_i\theta^{true} + E_i$ (capitals: RVs)

where the distribution of E_i is known, eg

$$E_i \sim N(0, \sigma^2), \quad E = [E_1, E_2, \ldots]^T$$

→ although we don't know E_i, we hence know the distribution of $\|E\|^2$ from the distribution of E_i →

Chi-squared distribution

Give a series of standard normal random variables, ie $E_i \sim N(0,1)$ ('standard')

The distribution of

$$Q = \sum_{i=1}^{k} E_i^2$$

ie sum of squared errors

is called the chi-square / chi-squared distribution with k 'degrees of freedom' (number of errors).

Write $Q \sim \chi_k$

If we assume $Y_i = X_i\theta^{true} + \varepsilon_i$ for $i = 1, \dots, m$

where $\varepsilon_i = N(0, \sigma^2)$ (σ^2: variance), then

$E_i = \frac{Y_i - X_i\theta^{true}}{\sigma}$ have dist. $N(0,1)$ so (σ: standard deviation)

$$\Rightarrow \frac{1}{\sigma^2}\| Y - X\theta^{true} \|^2 \sim \chi_m$$

Y, X random vars here

Hypothesis test inversion & confidence intervals

We don't know θ^{true} but we can test

$$H_0 : \theta^{true} = \theta$$

for any θ!

Logic (Fisher):

- Suppose data is/are a 'typical' realisation, y^{obs}, such that the realised sum of squared errors lies in the 95% region of $\frac{1}{\sigma^2}\|Y - X\theta_{true}\|^2$
- If $\frac{1}{\sigma^2}\|y^{obs} - X\theta\|^2$ does not lie in the 95% region of $\frac{1}{\sigma^2}\|Y - X\theta\|^2$ then θ cannot be equal to θ^{true}
  - → rule it out!
  - → otherwise call non-rejected
    - ↳ not the same as 'accepted' as unique answer: ill-posed!
- Can show set of all non-rejected parameters forms a confidence set for θ^{true}

↳ ie a procedure for constructing random sets that will trap true value 95% of the time

Simplest acceptable fit models

The previous procedure gives us a set of non-rejected (compatible) models.

↳ 'Lack of uniqueness solved' by accepting a set of models

We might stop there! But can also ask for a single representative

→ (in general there will be no 'most complicated' model consistent with the data (unbounded complexity)

→ there often is a 'simplest' (in the sense of our regularisation) model though!

↳ gives a lower bound on complexity (Donoho)

Popper!

- ↳ simple models = falsifiable
  - ↳ (can be ruled out)
- ↳ complex models = non-falsifiable, though may be true!

Also Vapnik...

Procedure:

- Given m data points
- Find 95% upper limit for χ^2_m distribution (max misfit)
- Find min complexity (regularisation) model fitting no worse than above

Can also eg use 50% (median) as a more 'central' point estimate

↳ more typically used approach.

↳ convenient fact: for large m, the median of χ^2_m is approx m.

→ choose $\|y^{obs} - X\theta\|^2 \approx \sigma^2 \cdot m$

for 'median' discrep. model.

Discrepancy approaches

Pros

- statistically well-justified } as interval method
- simple interpretation

Cons

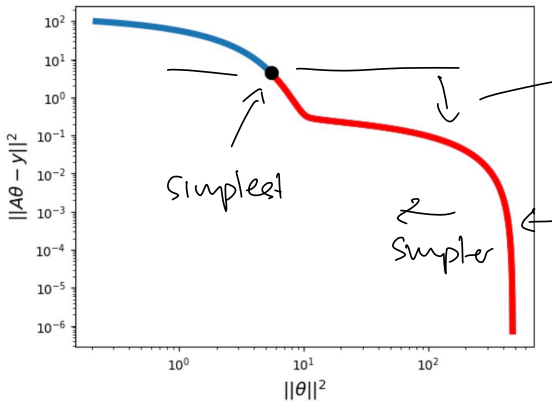
- need to know σ or noise model in general
- tends to oversmooth when used as a point estimate (remember it's a lower bound!)
  - even when taking median too.

Example: polynomial ($x^{[4]}$) regression, known noise level ($\sigma = 1$)

- Tikhonov $\lambda\|\theta\|^2$
- Loop over each λ, calc norms
- Then:

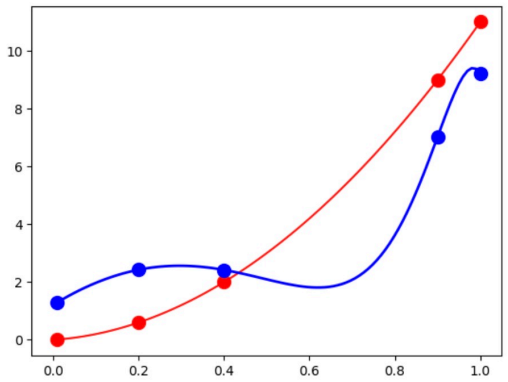
median

```
from scipy.stats import chi2
delta = sigma**2*chi2.ppf(q=0.5,df=len(yp_obs))
indices = data_norms**2 <= delta
data_norms_delta = data_norms[indices]
theta_norms_delta = theta_norms[indices]
nreg = len(data_norms_delta)

plt.loglog(theta_norms**2,data_norms**2,linewidth=5)
plt.loglog(theta_norms_delta**2,data_norms_delta**2,'r',linewidth=5)
plt.loglog(theta_norms[nreg]**2,data_norms[nreg]**2,marker='o',markersize=10,linewidth=5,color='k')

plt.ylabel(r'$||A\theta-y||^2$',fontsize=14)
plt.xlabel(r'$||\theta||^2$',fontsize=14)
plt.show()
```

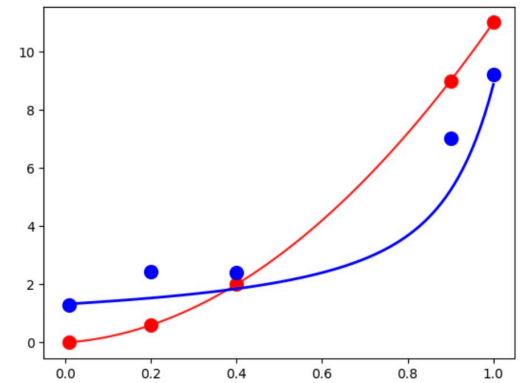


Note: even median quite oversmoothed as point estimate

Example cont'd



pseudo inverse sol (blue)



discrepancy principle (over smoothed but treat as lower bound)

Cross-validation & prediction

Cross-validation & its many variants are also widely used to choose regularisation parameters in inverse problems, statistics & machine learning

As before we aim to 'correct' the naive approach of just considering the models that just fit the given — here, 'training' — data

Why? Firstly, $\|y^{obs} - X\theta^{fit}\|^2$, where θ^{fit} is the model from minimising $\|y^{obs} - X\theta\|^2$ over θ, is known / can be shown to be a biased estimate of

$$\boxed{Err = \mathbb{E}\left[\|Y - X\theta^{true}\|^2\right]}$$

(can easily generalise to nonlinear)

which is the expected prediction error, where θ^{true} is fixed, Y is randomly drawn from the distribution indexed by θ^{true}, X is sometimes random, typically fixed, ie the expected error for repeatedly predicting new samples at given X values using the true model.

Prediction cont'd

Why care about prediction?

1. intrinsically reasonable thing to care about!

2. It can be shown that θ^{true} minimises the expected error for samples drawn from the distribution associated with θ^{true} – as it should be! ie

$$\underset{\theta}{\arg\min}\ Err(\theta) = \mathbb{E}_{y|x;\theta^{true}}\left[(y - X\theta)^2\right]$$

$$= \theta^{true}$$

→ this is a form of statistical 'consistency' property.

But we don't know θ^{true}! And we only have finite data available: $\hat{E}$ rather than $\mathbb{E}$

Bias & variance

Although $\|y^{obs} - X\theta^{fit}\|^2$ where θ^{fit} is a model fit using y^{obs} (perhaps via least sq.) is typically biased, it may have low variance.

→ In statistics, $\boxed{\text{risk} = \text{bias}^2 + \text{variance}}$ ← quadratic loss

& we may accept some bias.

However, if the model family is very large, eg so that we can exactly fit any data, the bias is very large

↳ always estimate error = 0!

→ hence we may want to construct an (approximately) unbiased estimate of the prediction error for any model, really any model fitting strategy

→ this motivates cross-validation

↳ more sophisticated versions may trade some bias & variance of pred. error estimate!

Recipe: leave-one-out cross validation

Given a sample $(x, y)_{i=1,\dots,m}$

& a rule for fitting a model to data, $\theta_\lambda^{fit}((x,y)_{i=1,\dots m})$ where λ controls the allowed model complexity, eg

$$\theta_\lambda^{fit} = \arg\min_\theta \|y^{obs} - X\theta\|^2 + \lambda\|L\theta\|^2$$

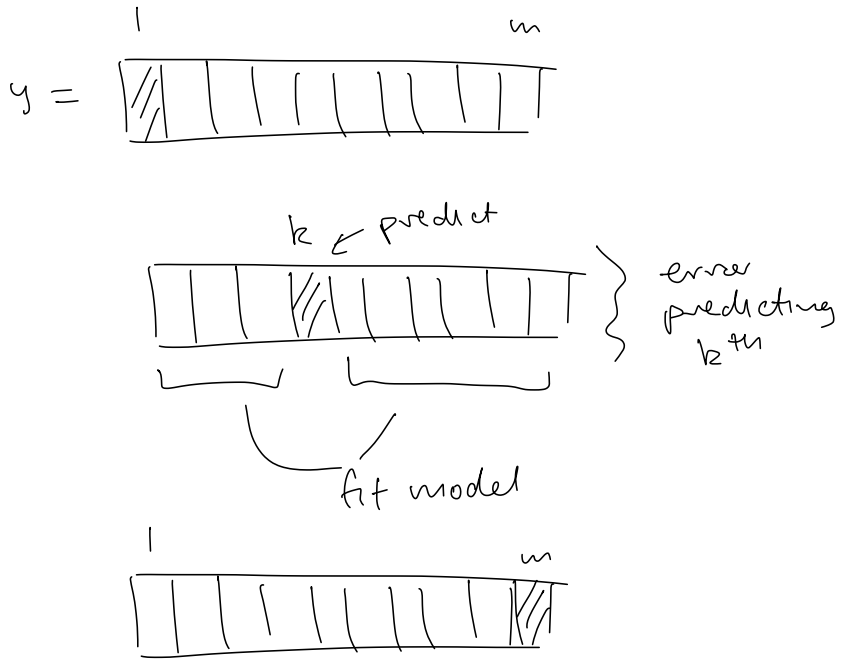
we define $\theta_\lambda^{fit, i\neq k}$ as the model fit on all data except the k^{th} pair $(x,y)_k$ & $X_{[k,:]}$ as the observed forward map for the k^{th} observation

& estimate the prediction error of the fitting rule by

$$\boxed{err^{loocv}(\theta_\lambda^{fit}) = \frac{1}{m}\sum_{k=1}^{m}\left(y_k^{obs} - X_{[k,:]}\theta_\lambda^{fit, i\neq k}\right)}$$

ie the average error from fitting the model on the data with the k^{th} data point removed & predicting the k^{th} data point, for $k = 1, \dots m$ ie each data point in turn.

Picture:



average error over $k = 1, \dots m$

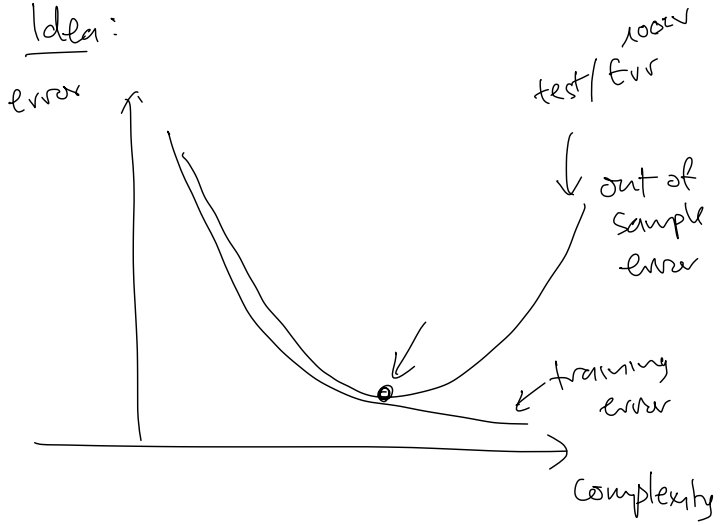
Estimating λ

Given our estimates for each λ, we choose the λ that minimizes

$$\text{Err}^{\text{loocv}}\left(\hat{\theta}^{\text{fit}}_{\lambda}\right) = \text{Err}^{\text{loocv}}(\lambda)$$

($\hat{\theta}^{\text{fit}}_{\lambda}$: only depends on λ)

Idea:



→ typically refit on whole dataset with chosen λ to get final θ estimate (this can cause issues!)

Notes

- Simple & general, use for any fitting rule
- unbiased estimate of average predictive error for $n-1$ data points, $\approx$ average predictive error for n data points
- averages over both inputs & outputs → error rate for X random rather than fixed (ie not error rate for this sample but for rule in general)
- high variance estimate of prediction error!
  - ↳ can improve with eg k-fold cross validation or bootstrap versions

K-fold: leave out $\approx n/K$ observations at a time instead of one → lower variance, higher bias (typically...)

- computationally expensive! K fits per λ!

Generalised cross validation : cheaper (for linear)

For linear models with linear regularisation, i.e. can be written

$$y^{pred} = X X^{\#}_{\lambda} y^{obs}, \quad X^{\#}_{\lambda} : \text{'regularised inverse'}$$

$$= R^{\lambda}_{d}\, y^{obs} \quad \left\{ \begin{array}{l} \text{data space} \\ \text{pseudo projector} \end{array} \right.$$

can show a reasonable approximation to $Err^{loocv}(\lambda)$ is

$$\frac{1}{m} \frac{\| y^{obs} - X\theta^{fit}_{\lambda} \|^2}{\left(1 - \frac{\text{Trace}(X X^{\#}_{\lambda})}{m}\right)^2} = \frac{1}{m} \frac{\| y^{obs} - X\theta^{fit}_{\lambda} \|^2}{\left(1 - \frac{\text{Trace}(R_d)}{m}\right)^2}$$

correction factor

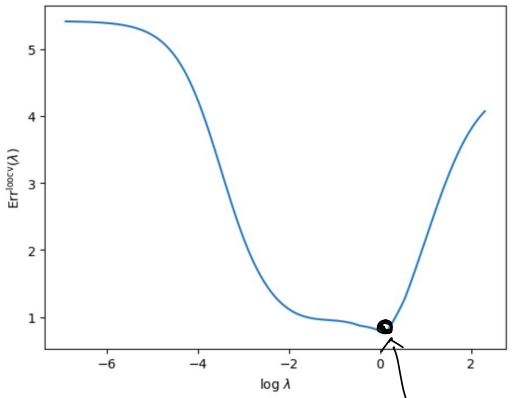
like eg $\sigma^2 = \frac{1}{n-1} \Sigma (y_i - \mu)^2$ for unbiased variance!

[Taylor series approx. leads to $\approx$ AIC]

See Aster et al Inverse problems
Hastie et al. The elements of statistical learning.

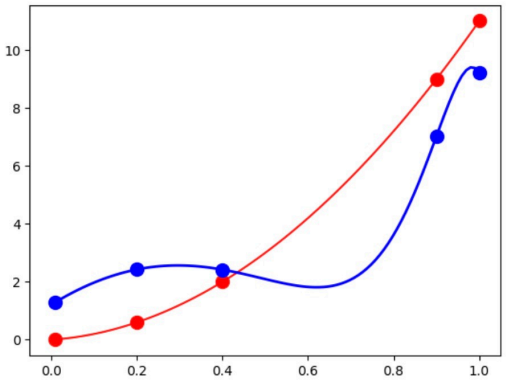
Example : CV attempt

```
thetas = np.zeros((len(alphav),len(thetap_true)))
theta_norms = np.zeros((len(alphav),len(yp_obs)))
data_norms = np.zeros((len(alphav),len(yp_obs)))
L = np.eye(len(thetap_true))
indices = np.arange(0,len(yp_obs),dtype=int)

for i, alphai in enumerate(alphav):
    for k in range(0,len(yp_obs)):
        # indices
        keep_indices = indices[indices != k]

        #augment
        A_poly_aug_k = np.vstack([A_poly_obs[keep_indices,:],alphai*L])
        yp_aug_k = np.hstack([yp_obs[keep_indices],np.zeros(L.shape[0])])

        #invert
        A_poly_aug_k_pinv = np.linalg.pinv(A_poly_aug_k)
        theta_k = np.dot(A_poly_aug_k_pinv,yp_aug_k)

        #calc norms: note prediction on k!
        theta_norms[i,k] = np.linalg.norm(L@theta_k,2)
        data_norms[i,k] = np.linalg.norm(yp_obs[k] - np.dot(A_poly_obs[k,:],theta_k))

# average data errors over k for each lambda
data_norms_cv = np.mean(data_norms,axis=1)
```



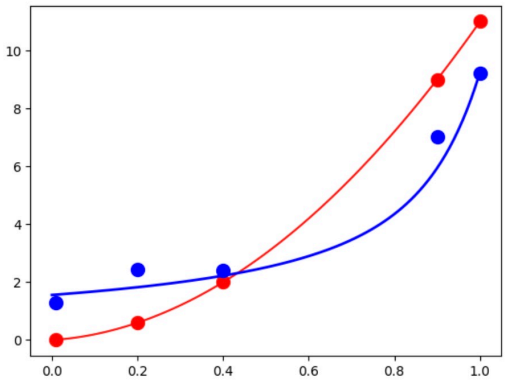
min $\lambda \approx 1$

Example cont'd



no regularisation

(min norm A^+)



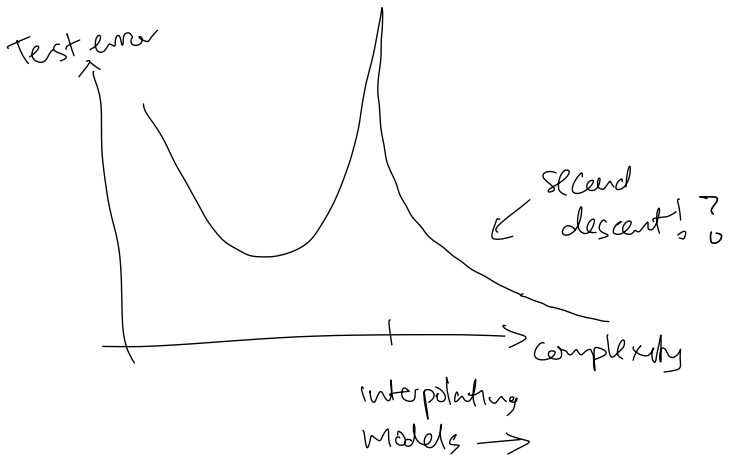
CV + Tikhonov

Further reading

- Aster et al Inverse Problems
- Hastie et al The elements of Statistical learning
- Efron (1983) 'Estimating the error rate of a prediction rule'
- Bates et al (2024) 'Cross-validation: what does it estimate & how well does it do it?'

Appendix : Double descent??

It has been observed that we can sometimes get plots like



However, imo they usually result from plotting vs the wrong measure of complexity!

} does demonstrate that interpolating models can generalise through!

→ see Assignment 2??

→ Curth et al (2024) "A U-turn on double descent: rethinking parameter counting in statistical learning"

etc!,