Biomeng 261 Lecture 6: $\boxed{\text{Intro to Parameter Estimation}}$

Usual modelling procedure:

parameters → $\boxed{\text{model}}$ → predictions
initial conditions ↗
etc

eg

$k_1, k_{-1}, k_2, \ldots \rightarrow \boxed{\text{ODEs}} \rightarrow [S](t)$
$[S](0), [E](0), \ldots \nearrow$ $\qquad [E](t)$
$\vdots$

! $\boxed{\text{Problem}}$: we usually have data (ie outputs) & don't know exact inputs eg k_1, k_{-1} etc

input → $\boxed{\text{model}}$ → {outputs}

↑ wanted

? ←

(inverse problem/ estimation problem/ fitting problem)

↑ measured/ known

'Well-posed' problems (Hadamard)

The solution
- exists
- is unique
- is stable (under variations in given data) (since not known exactly)

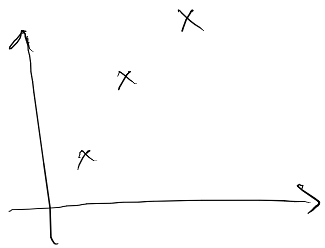
! $\boxed{\text{Estimation}}$ is an ill-posed problem
- might not be even one good fit among models considered ('misspecified')
- might be many solutions ('unidentifiable')
- even a unique solution might be unstable if data is varied a little ('overfit')

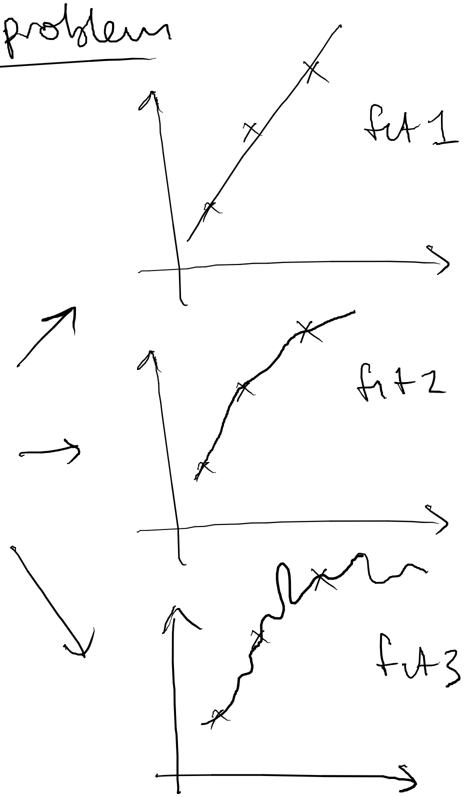
Moral: $\boxed{\text{trade-offs}}$ between
- acceptable fit to given data
- class/complexity of models considered
- etc! Be pragmatic & skeptical

Illustration:

Curve fitting problem

given data





Which is better?

Finite amount of data

⇒

many compatible models

More complex fits better

→ is 'fit' always better?

→ often multiple, conflicting goals

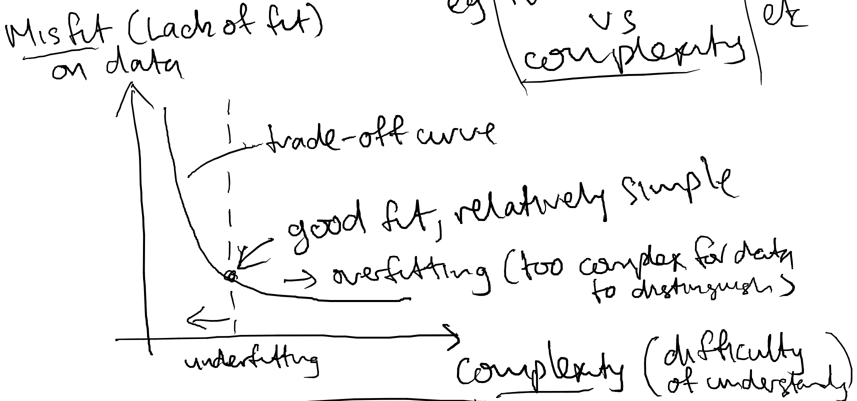
Note

Often: simpler ≈ more understandable

– BUT: fits less well!

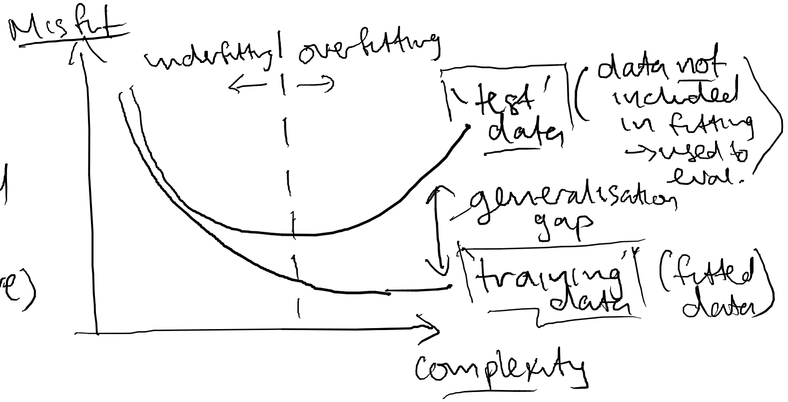
Trade-offs are key } you need to decide:

eg fit to data vs complexity etc



also has purely predictive justification:

(Machine/ Statistical learning perspective)



Here: Split full data → training data
→ test data

Measuring data fit/misfit

Need a $\boxed{\text{distance}}$ function (ie $\boxed{\text{metric}}$):

$$\boxed{d(\underline{y}_d, \underline{y}_m)}$$

[$\underline{y}$: vector
eg $\underline{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \end{pmatrix}$]

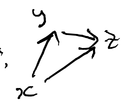
y_d : actual, measured/exp. data

y_m : model data (eg simulated)

Properties (metrics – note: between pairs)

1. $d(x,y) = d(y,x)$ symmetric
2. $d(x,y) \geq 0$ non-negative
3. $d(x,y) = 0 \iff x = y$
4. $d(x,z) \leq d(x,y) + d(y,z)$

↳ Triangle inequality:



semi/pseudo metric if replace 3. by 3'. $\boxed{d(x,x) = 0}$ but poss. $d(x,y) = 0$ & $x \neq y$

The 'size' or $\boxed{\text{norm}}$ of a $\boxed{\text{single}}$ object or vector can then be defined by

$$\boxed{\|x\|_d = d(x,0)}$$

– ie distance from zero

Optimal trade-off solutions (Pareto)

1. minimise [model complexity] ← objective function

subject to acceptable data fit ← constraint

$$\boxed{\begin{aligned} &\min_{\theta}\ d_1(\theta, 0) \leftarrow \text{parameter space metric} \\ &\text{s.t. } d_2(y(\theta), y_d) \leq \delta \text{ — tolerance} \end{aligned}}$$

parameters (vector)

↑ model prediction of data for given θ

↑ measured data.

---

2. minimise [data misfit]

s.t. acceptable complexity

$$\boxed{\begin{aligned} &\min_{\theta}\ d_2(y(\theta), y_d) \\ &\text{s.t. } d_1(\theta, 0) \leq \epsilon \end{aligned}}$$

---

3. minimise [data misfit + weighted complexity penalty]

convenient since single 'objective function' →

$$\boxed{\min_{\theta}\ d_2(y(\theta), y_d) + \lambda d_1(\theta, 0)}$$

weight/conversion (different distance scales)

Each form has a 'tuning' / 'trade-off' or 'hyper' parameter

eg $\delta, \epsilon, \lambda$

- Can show that the three formulations are equivalent given right choice of $\delta, \epsilon, \lambda$ (can convert between)

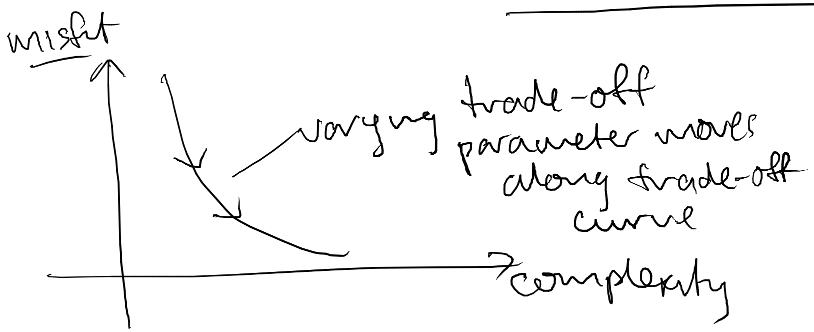
$\min d_2 + \lambda d_1$

prob. easiest tho'.

- We usually vary the trade-off parameter(s) to get the whole trade-off curve

if possible: depending on resources

- chose δ
- solve problem
- choose new δ etc



∘ each point on curve is 'best' for choice of trade-off parameter ('Pareto efficient')

Typical distances

$$d(\underline{x}, \underline{y}) = \sqrt{\sum_{i}^{n} (x_i - y_i)^2}$$

[sum of squares]

$$d(\underline{x}, \underline{y}) = \sum_{i}^{n} |x_i - y_i|$$

[sum of absolute differences]

⇒ can mix & match data & param. distances

→ different 'robustness' & 'efficiency' properties

another trade-off!

↳ least squares widely used & 'best' under Gaussian noise/error but sensitive to outliers

↳ absolute differences less sensitive to outliers, but also less sensitive to true differences

---

.... Finally: Let's apply!