Boring but important disclaimers:

- If you are not getting this from the GitHub repository or the associated Canvas page (e.g. CourseHero, Chegg etc.), you are probably getting the substandard version of these slides Don't pay money for those, because you can get the most updated version for free at

  `https://github.com/julianmak/academic-notes`

  The repository principally contains the compiled products rather than the source for size reasons.
- Associated Python code (as Jupyter notebooks mostly) will be held on the same repository. The source data however might be big, so I am going to be naughty and possibly just refer you to where you might get the data if that is the case (e.g. JRA-55 data). I know I should make properly reproducible binders etc., but I didn't...
- I do not claim the compiled products and/or code are completely mistake free (e.g. I know I don't write Pythonic code). Use the material however you like, but use it at your own risk.
- As said on the repository, I have tried to honestly use content that is self made, open source or explicitly open for fair use, and citations should be there. If however you are the copyright holder and you want the material taken down, please flag up the issue accordingly and I will happily try and swap out the relevant material.

OCES 3301 :

basic Data Analysis in ocean sciences

Session 4: regression

Outline

(Just overview here; for actual content see Jupyter notebooks)

- multi-linear regression
- measurement of skill and complexity
- principal component analysis (PCA)

Recall: linear regression

Figure: The eternal bendy boi.

Multi-linear regression

Beyond basic linear regression, I could go higher degree,

$$y = g(x) = a_0 x^n + a_1 x^{n-1} + \cdots a_{n-1} x + a_n = \sum_{i=0}^{n} a_i x^{n-i},$$

Multi-linear regression

Beyond basic linear regression, I could go higher degree,

$$y = g(x) = a_0 x^n + a_1 x^{n-1} + \cdots a_{n-1} x + a_n = \sum_{i=0}^{n} a_i x^{n-i},$$

but I could also keep it linear and go beyond single variable,

$$y = g(x_1, x_2, \ldots) = a_0 + a_1 x_1 + a_2 x_2 + a_n x_n = a_0 + \sum_{j=1}^{n} a_j x_j,$$

where x_j here denotes a **different input variable**, so the samples within the variable x_j might be denoted $x_{i,j}$

Multi-linear regression

e.g.

$$\begin{aligned}\text{cursedness} = {} & a \times \text{size} \\ & + b \times \text{distortion} \\ & + c \times \text{Eldritch attribute} \\ & + d \times \text{colour} + \dots\end{aligned}$$

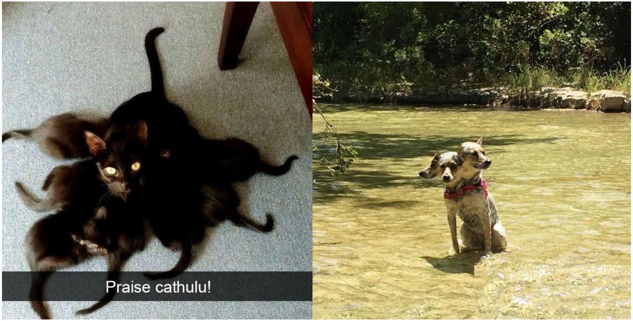

Figure: Which is more cursed?

Skill vs. complexity

Is a model with lower mismatch necessarily better?

- overfitting?
- substantially increased complexity with small gain in mismatch?

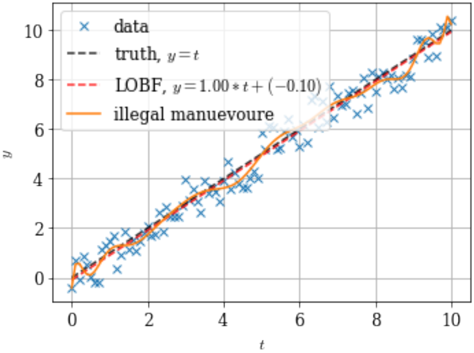

Figure: Linear regression example.

Skill vs. complexity

A measure to reward reduction in mismatch and peanlise complexity

- AIC (Aikake Information Criterion)
- BIC (Bayesian Information Criterion)

  → sometimes **Schwarz Information Criterion**

- **lower** A/BIC values are "good"

  → only a relative measure

  → like-for-like comparison with model trained from **same** data

  → BIC penalises complexity more than AIC

PCA

Principal Component Analysis (PCA)

- picks out the "most important" features from data set

  → measured through variance explained

  → pulls out the PCs that are uncorrelated to each other
- can be useful in exploratory data analysis
- lower dimensional reduction

  → use for visualising high dimensional data in a sense
- filtering out noisy data

- will visit again in the form of EOFs

  → may visit again in the machine learning session if it happens

PCA

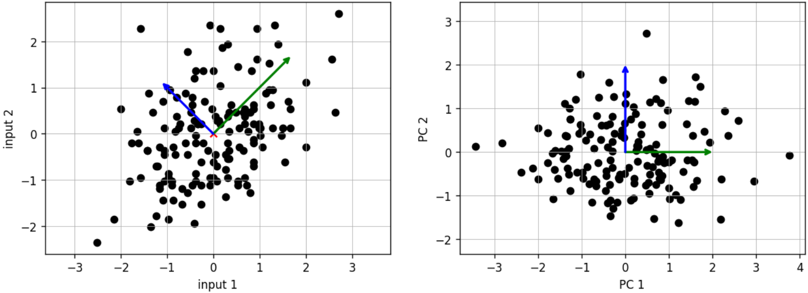

Figure: PCA of sample 2d penguin data, picking out the PCs in input co-ordinates (left), and in PC co-ordinates (right).

PCA

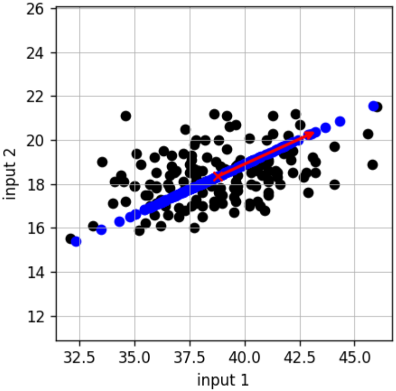

Figure: PCA of sample 2d penguin data, picking out only the first PC, project data onto PC1, and transform it back into input co-ordinates.

PCA

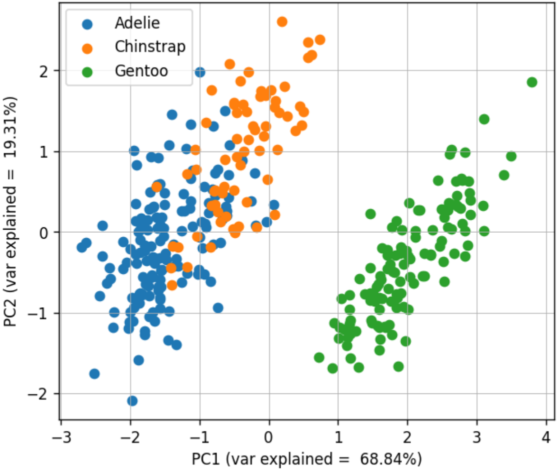

Figure: PCA example for full penguin data, showing a 3d section of the 4d data in input co-ordinates (left), and the full 4d data projected onto the PC1 and PC2, given in PC co-rodinates.

PCA: more elaborate examples

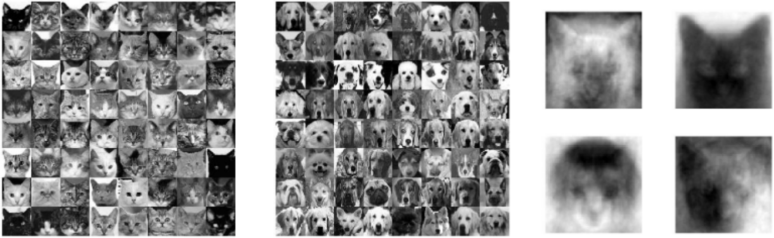

Figure: PCA example on cats and dogs, showing the first 4 PCs (cf. the arrows in the graphs before). Figure adapted from Fig. 10 of Brunton, Brunton, Proctor & Kutz (2013).

- search for `eigenfaces` if you want some more stuff of nightmares

Jupyter notebook

Go to 04 Jupyter notebook to play around with the iris and penguin data

- the example in notebook explains PCA in more detail and a bit slower

  → will visit again in the form of EOFs
- may be more of this in a machine learning extra session (if it happens)