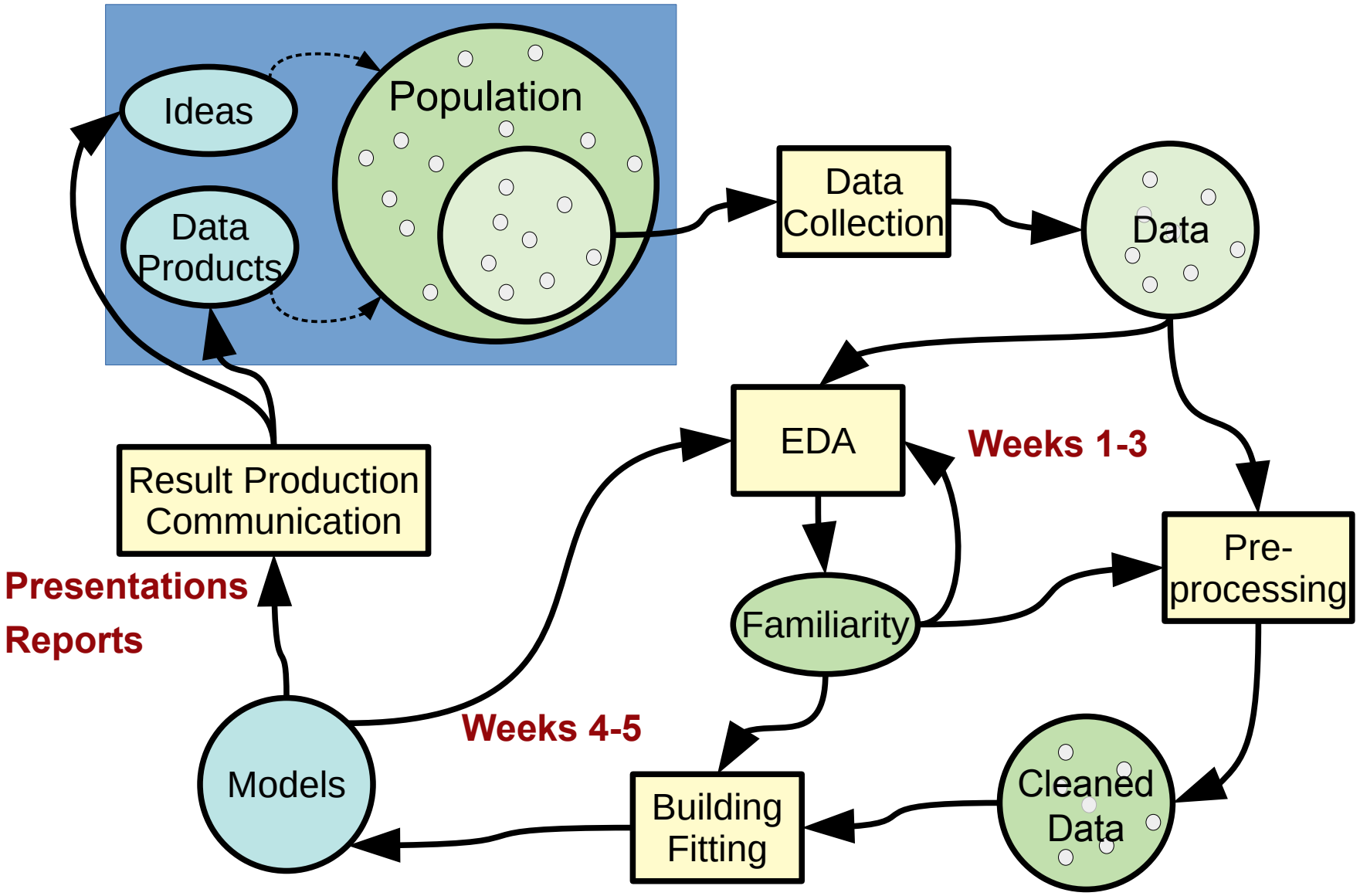
Course Content



Application of numerical data description

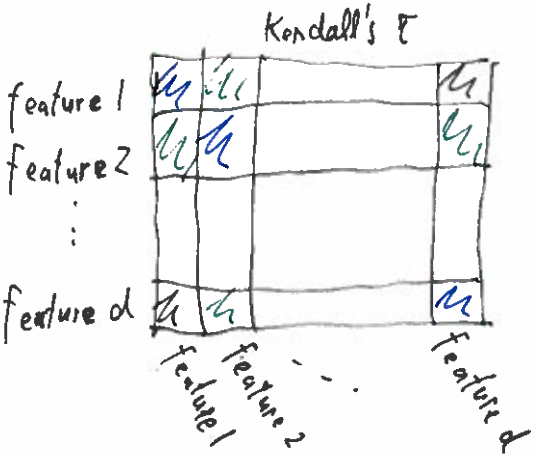
- Cheap ⇝ apply plenty (robust and non-robust)
- Report in a meaningful way: take selection bias into account
- Understand structure of data

Do this on training set

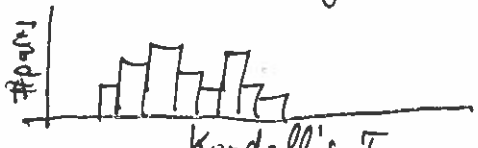
First step in data analysis:
Set aside held-out set

Then understand structure of remaining data.

Pairwise measures:
- Visualise matrix



- Visualise histogram



- Minimise approximation error

$$\underset{w_1,\dots,w_k}{\text{minimise}} \; \mathbb{E}\left[\left\|\underline{x} - \sum_{i=1}^{k} \underline{w}_i \underline{w}_i^T \underline{x}\right\|^2\right]$$

$$\text{s.t. } \|\underline{w}_i\| = 1, \; i = 1,\dots,k \text{ and } \underline{w}_i^T \underline{w}_j = 0$$

- Approximate data matrix X

$$\underset{M}{\text{minimise}} \sum_{i,j} (X - M)_{ij}^2 \quad \text{s.t. } \operatorname{rank}(M) = k$$

$$M = \sum_{i=1}^{k} s_i \underline{u}_i \underline{v}_i^T$$

$$\underline{w}_i = \underline{u}_i \; ; \; z_{i:} = s_i \underline{v}_i^T$$

- Approximate sample covariance

$$\underset{M}{\text{minimise}} \; \left\|\frac{1}{n} X X^T - M\right\|_F \quad \text{s.t. } \operatorname{rank}(M) = k \text{ and } M^T = M$$

$$\hat{\Sigma} = \sum_{i=1}^{k} \underline{u}_i \frac{s_i^2}{n} \underline{u}_i^T$$

- Approximation Gram matrix

$$\underset{M}{\text{minimise}} \; \|X^T X - M\|_F \quad \text{s.t. } \operatorname{rank}(M) = k \text{ and } M^T = M$$

$$\hat{G} = \sum_{i=1}^{k} \underline{v}_i s_i^2 \underline{v}_i^T$$

$$Z = \sqrt{\tilde{\Lambda}_k}\, V_k^T$$

- Kernel PCA:

Define $\tilde{G}$ with $(\tilde{G})_{ij} = \phi(\underline{x}_i)^T \phi(\underline{x}_j)$

$= k(\underline{x}_i, \underline{x}_j)$

- Classical MDS:

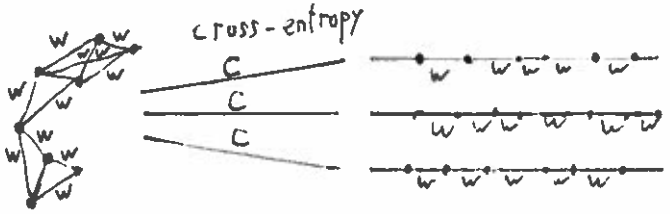
$$Z = \underset{M}{\text{minimise}} \left\| \left(-\tfrac{1}{2} C_n \Delta C_n\right) - M^T M \right\|_F$$

s.t. $\text{rank}(M^T M) = k$

- Isomap:

Classical MDS, but on geodesic distances

- UMAP:



$\tilde{X}$ Potential Z

$$\underset{Z}{\text{minimise}}\ C(W^{(\tilde{X})}, W^{(Z)})$$

Estimating generalisation performance with hyperparameter selection

1. Split D into D^{hyp} and D^{test} (e.g. 80:20)

   Lock away D^{test} until step 4

2. Optimise hyperparameters $\underline{\lambda}$ on D^{hyp}:

held-out

Split D^{hyp} into D^{train} and D^{val}

$\hat{\underline{\lambda}} = \underset{\underline{\lambda}}{\operatorname{argmin}}\ PL(\underline{\lambda})$

where

$PL(\underline{\lambda}) = \hat{J}(\hat{h}_{\underline{\lambda}}, D^{val})$

and $\hat{h}_{\underline{\lambda}} = \mathcal{A}_{\underline{\lambda}}(D^{train})$

cross validation

Split D^{hyp} into K folds and compute CV for each $\underline{\lambda}$

$\hat{\underline{\lambda}} = \underset{\underline{\lambda}}{\operatorname{argmin}}\ EPL(\underline{\lambda})$

where

$EPL(\underline{\lambda}) = CV_{\underline{\lambda}} = \hat{J}(\mathcal{A}_{\underline{\lambda}})$

3. Reestimate $\underline{\theta}$ using $\hat{\underline{\lambda}}$: $\hat{h}^{hyp} = \mathcal{A}_{\hat{\underline{\lambda}}}(D^{hyp})$

4. Estimate prediction loss: $\hat{J}(\hat{h}^{hyp}; D^{test})$

5. Reestimate $\hat{h}$ on all data: $\hat{h} = \mathcal{A}_{\hat{\underline{\lambda}}}(D)$

Do not estimate prediction loss on all data!

"Two-times held-out" "CV and held-out"

Loss functions

- Regression:
  square loss, absolute loss, Huber loss

- Classification
  - loss matrix, zero-one loss
    → true positive rate, true negative rate,
    false positive rate, false negative rate
  - ROC: true positive rate vs. false positive rate
  - differentiable loss functions
    based on margin $yh(\underline{x})$