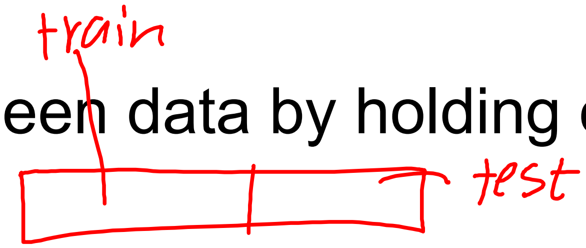
When Models Meet Data 2

Liang Zheng
Australian National University
liang.zheng@anu.edu.au

Overfitting

- The aim of a machine learning predictor is to perform well on unseen data.
- We simulate the unseen data by holding out a proportion of the whole dataset.



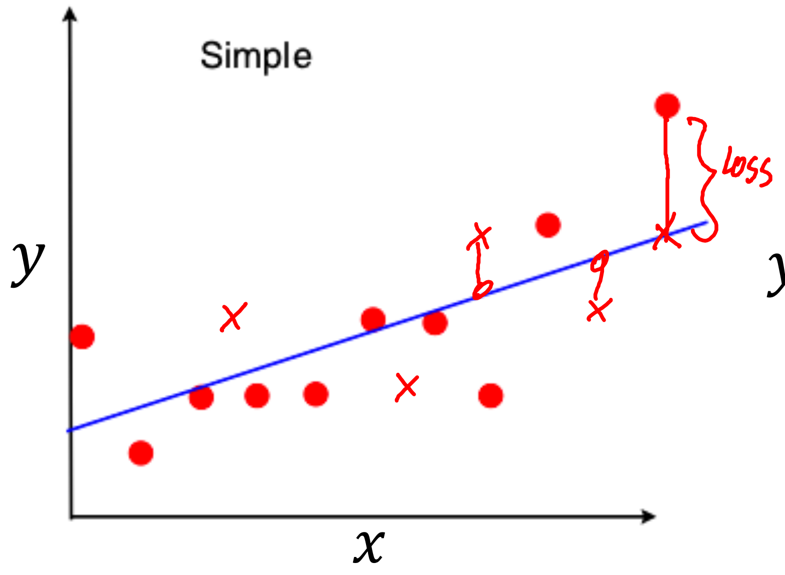
- This hold out set is called test set.

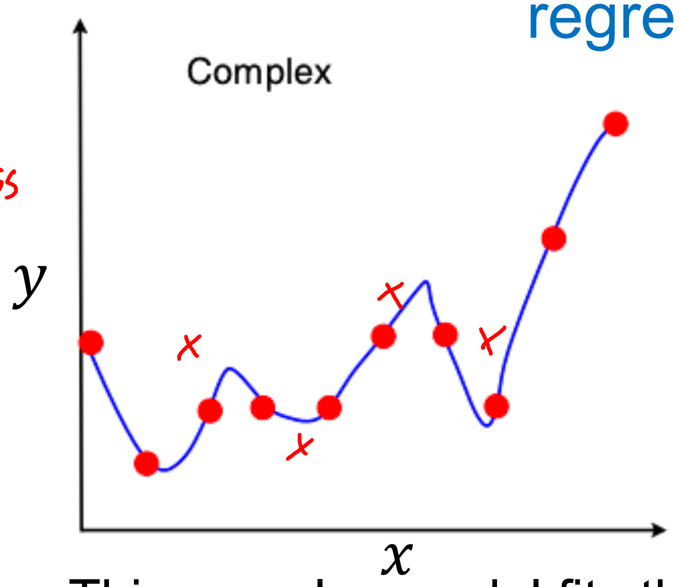
- In practice, we split data into a training set and a test set.
- Training set: fit the model
- Test set: not seen during training, used to evaluate generalization performance

- It is important for the user to not cycle back to a new round of training after having observed the test set.

- Empirical risk minimization can lead to overfitting.
  = average loss
- the predictor fits too closely to the training data and does not generalize well to new data

regression





This simple model fits the training data less well.

A larger empirical risk.

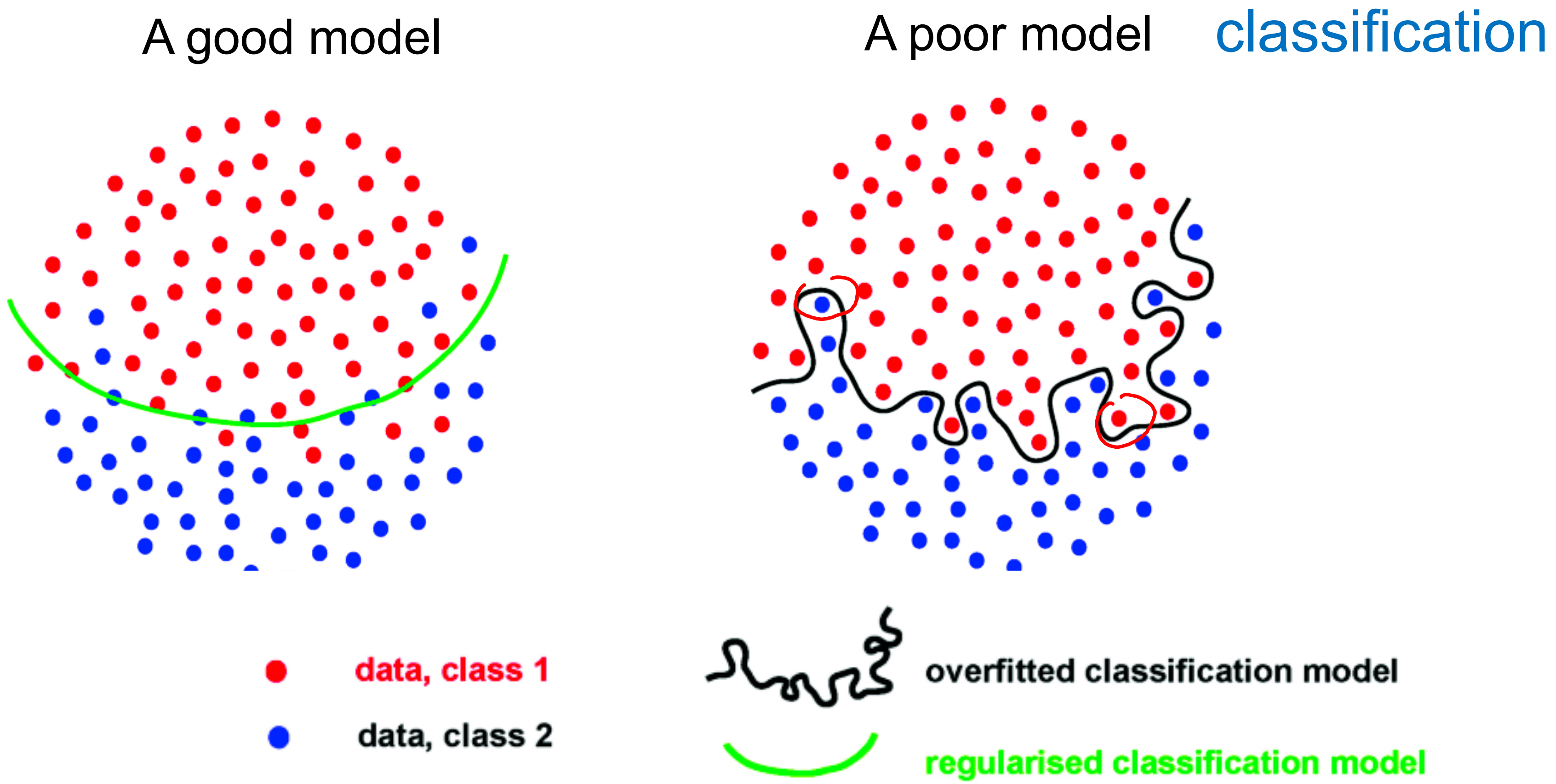
A good machine learning model.

This complex model fits the training data very well.

A very small empirical risk.

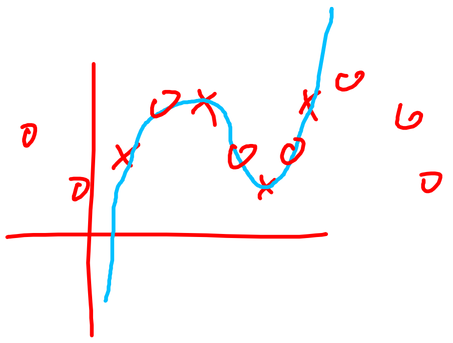
A poor machine learning model due to overfitting.

- Empirical risk minimization can lead to overfitting.
- the predictor fits too closely to the training data and does not generalize well to new data

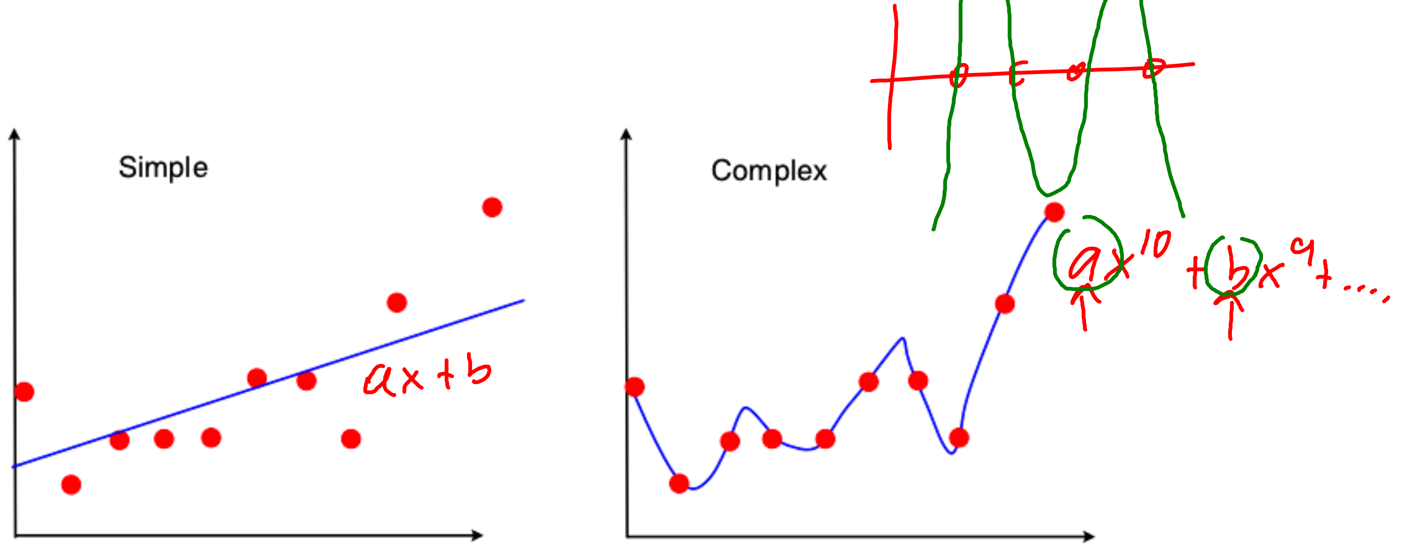


8.2.3 Regularization to Reduce Overfitting

- When overfitting happens, we have
  - very $\overset{\text{zero}}{\text{small}}$ average loss on the training set but large average loss on the test set
- Given a predictor f, overfitting occurs when
  - the risk estimate from the training data $\mathbf{R}_{\text{emp}}(f, \boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ underestimates the expected risk $\mathbf{R}_{\text{true}}(f)$. In other words,
  - $\mathbf{R}_{\text{emp}}(f, \boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ is much smaller than $\mathbf{R}_{\text{true}}(f)$ which is estimated using $\mathbf{R}_{\text{emp}}(f, \boldsymbol{X}_{\text{test}}, \boldsymbol{y}_{\text{test}}) \approx R_{true}(f)$
- Overfitting occurs usually when
  - we have little data and a complex ~~hypothesis class~~ model



- How to prevent overfitting?
- We can bias the search for the minimizer of empirical risk by introducing a $\underbrace{\text{penalty term}}$
- The penalty term makes it harder for the optimizer to return an overly flexible predictor
- The penalty term is called regularization.
- Regularization is an approach that discourages complex or extreme solutions to an optimization problem.



$[\lambda_1, \lambda_2, \lambda_3]^T \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{bmatrix}$

big small

$\boxed{\theta_1 x^2 + \theta_2 x + \theta_3}$

- Example
- Least-squares problem

ground truth $\hat{y}$

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{X\theta}\|^2$$

$$\bar{\theta} = \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{bmatrix}$$

$y_1 \; x_1$
$\vdots$
$y_n \; x_n$

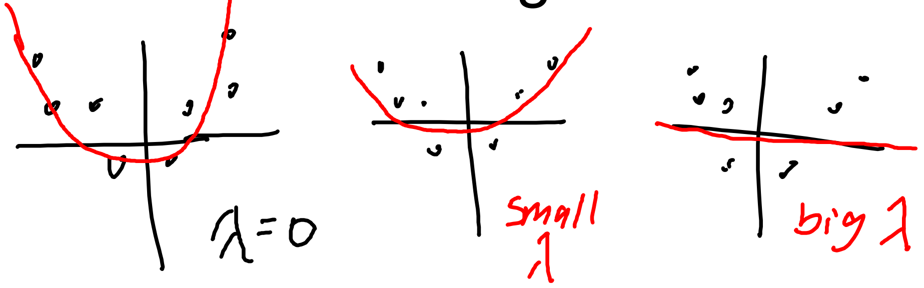
$\sin x_1$
$\sin x_2$

- To regularize this formulation, we add a penalty term

$$X = \begin{bmatrix} 1 & y_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & y_n & x_n^2 \end{bmatrix}$$

$$\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{N}\|\boldsymbol{y} - \boldsymbol{X\theta}\|^2}_{\text{Loss}} + \underbrace{\lambda\|\boldsymbol{\theta}\|^2}_{\text{regularizer}}$$

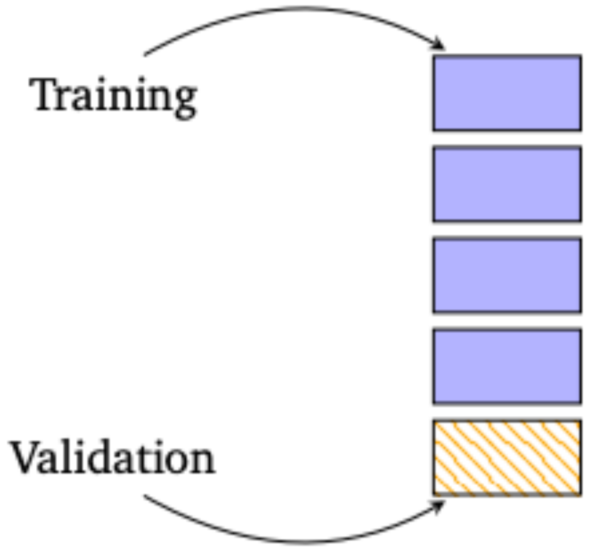
- The addition term $\|\boldsymbol{\theta}\|^2$ is called the regularizer or penalty term, and the parameter regularizer λ is the regularization parameter.
- λ enables a trade-off between minimizing the loss on the training set and the amplitude of the parameters $\boldsymbol{\theta}$
- It often happens that the amplitude of the parameters in $\boldsymbol{\theta}$ becomes relatively large if we run into overfitting
- $\boxed{\lambda \text{ is a hyperparameter}}$



8.2.4 Cross-Validation to Assess the Generalization Performance

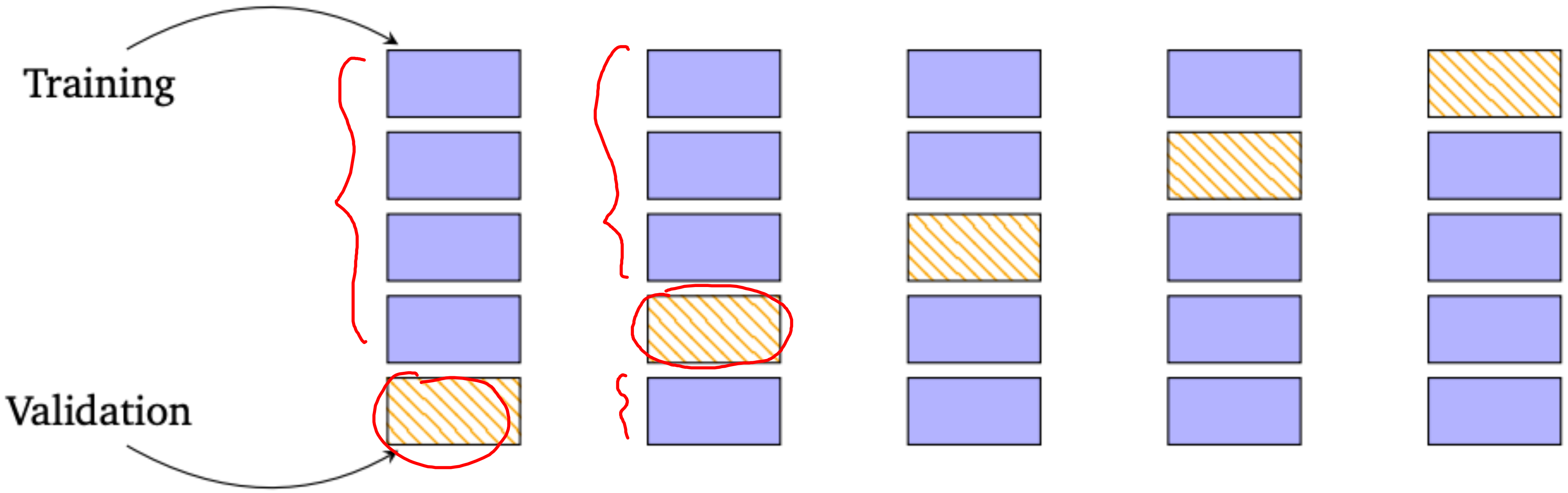
- We mentioned that we split a dataset into a training set and a test set
- we measure generalization error by applying the predictor on test data.
- This data is also sometimes referred to as the validation set.
- Validation set is from the entire data, and has no overlap with the training data.
- We want the training set to be large
- That leaves the validation set small
- A small validation set makes

 the result less stable (large variances)



- Basically, we want the training set to be large
- We want the validation to be large, too
- How to solve these contradictory objectives?
- Cross-validation: K-fold cross-validation $\equiv$ development

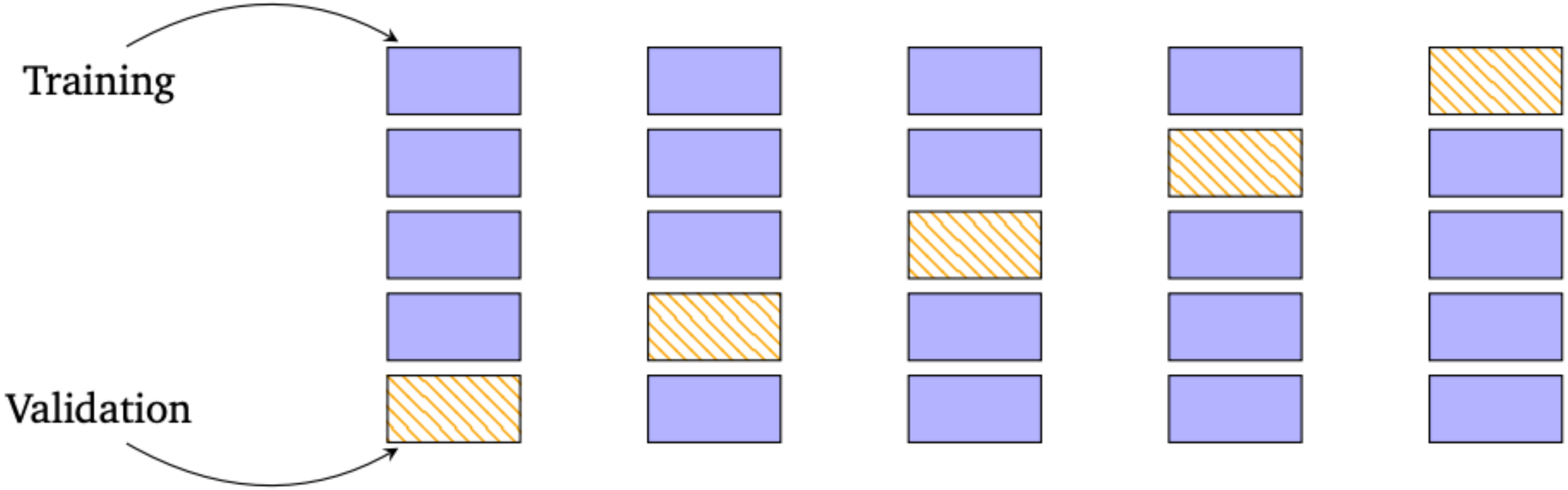
Example: $K = 5$



Cross-validation

- K-fold cross-validation partitions the data into K chunks
- $K-1$ trunks form the training set $\mathcal{R}$
- The last trunk is the validation set $\mathcal{V}$
- This procedure is repeated for all K choices for the validation set, and the performance of the model from the K runs is averaged

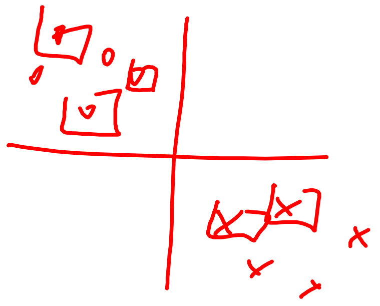
Example: $K = 5$



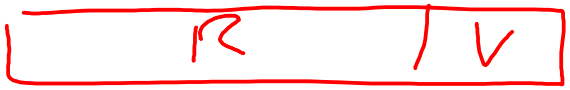
Cross-validation

- Formally, we partition our training set into two sets $\mathcal{D} = \underbrace{\mathcal{R} \cup \mathcal{V}}$, such that they do not overlap, i.e., $\mathcal{R} \cap \mathcal{V} = \phi$
- We train on our model on $\mathcal{R}$ (training set)
- We evaluate our model on $\mathcal{V}$ (validation set)
- We have K partitions. In each partition k:
  - Training set $\mathcal{R}^{(k)}$ produces a predictor $f^{(k)}$
  - $f^{(k)}$ is applied to validation set $\mathcal{V}^{(k)}$ to compute the empirical risk $\underbrace{\mathrm{R}\left(f^{(k)}, \mathcal{V}^{(k)}\right)}$
  - All the empirical risks are averaged to approximate the expected generalization error

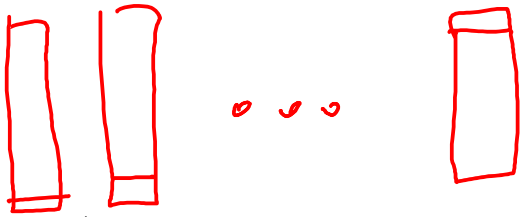
$$\mathbb{E}_{\mathcal{V}}[R(f, \mathcal{V})] \approx \underbrace{\frac{1}{K} \sum_{k=1}^{K} \mathrm{R}\left(f^{(k)}, \mathcal{V}^{(k)}\right)}$$



Cross-validation – some understandings



- The training set is limited -- not producing the best $f^{(k)}$
- The validation ~~testing~~ set is limited – may producing an inaccurate estimation of $\mathrm{R}\left(f^{(k)}, \mathcal{V}^{(k)}\right)$
- After averaging, the results are stable and indicative
- An extreme: leave-one-out cross-validation, where the validation set only contains one example.



- A potential drawback – computation cost
  - The training can be time-consuming
  - If the model has several parameters to tune, it is hard to evaluate those hyperparameters.
- This problem can be solved by parallel computing, given enough computational resources

Check your understanding

- T* When your model works poorly on the training set, your model will also work poorly on the test set. *most
- F* When your model works poorly on the training set, your model may also have overfitting. by def.
- T Overfitting happens when your model is too complex given your training data. by def
- F Regularization alleviates overfitting by improving the complexity of your training data. data is fixed
- T In K-fold cross-validation, we will get more stable test accuracy if K increases.
- T In 2-fold cross-validation, you can obtain 2 results from the 2 test sets, and they may differ a lot with each other.