Cross-validation

Mark van der Wilk

Department of Computing
Imperial College London

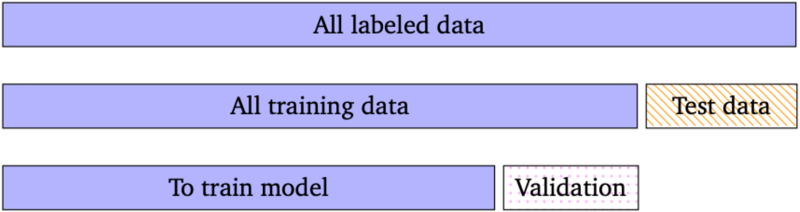
@markvanderwilk
m.vdwilk@imperial.ac.uk

November 7, 2022

Reading Material

Lecture notes, §8.1, §8.2
In particular §8.2.4 §8.6.1
https://mml-book.com

Recap: Validation set



- Use a **train set** to find the parameters in your model
- Use a **validation set** to select the model
- Keep test set **completely separate**, to use for estimating test performance

Parameter dependence on data

$$\mathbb{E}_{\prod_n \pi(x_n, y_n)}\left[\frac{1}{N}\sum_{n=1}^{N} \ell(f(\mathbf{x}_n; \boldsymbol{\theta}(\{\mathbf{x}_i, y_i\}_n)), y_n)\right] \neq \mathbb{E}_{p(\mathbf{x},y)}[\ell(f(\mathbf{x}; \boldsymbol{\theta}^*), y)]$$

param depends on data / param independent of data

- Goal: Estimate expected loss **for the parameter we pick**
- Expectation is not the same if parameter depends on the dataset
  $\implies$ **biased** estimate
- Not much more to the proof than that the equality does not hold.
- Remember example for insight:
  If we pick parameters for which loss is always zero
- Remember: we pick parameters with validation set
  $\implies$ need separate test set for unbiased estimation

Why is unbiasedness important?

Unbiasedness makes it easy to **prove** that we will end up with good estimates.

- Unbiasedness is helpful because we **only** need to control the **variance** of an estimate, to make it an accurate estimate of the expectation.
- Law of large numbers needs unbiasedness to be applied!
- The concentration inequalities we discussed needed unbiasedness!
- Biased estimators **may** be good, if you can control the bias. This may be difficult to verify.

If you come up with your own estimators: Just make them unbiased.

Unbiased estimation of expected loss

- Remember, we are interested in $\text{ER} = \mathbb{E}_{\pi(x,y)}[\ell(f(x;\boldsymbol{\theta}^*), y)]$.
- Prediction losses on a **separate set** of data are unbiased

$$L_{\text{test}} = \frac{1}{N}\sum_{n=1}^{N} \ell(f(x;\boldsymbol{\theta}^*)), y) \tag{1}$$

$$\implies \mathbb{E}_{\pi}[L_{\text{test}}] = \mathbb{E}_{\pi(x,y)}[\ell(f(x;\boldsymbol{\theta}^*), y)] \tag{2}$$

Test set size

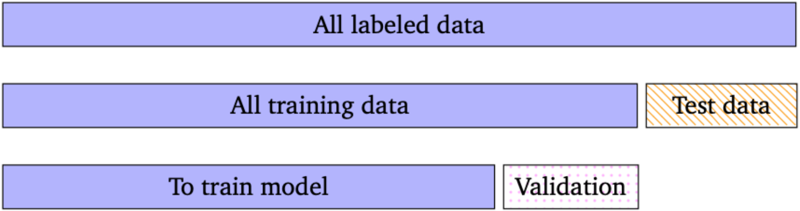
Property of test set: **unbiased** $\implies$ no systematic over/under estimation.
How many points to use in the test set?

$$\mathbb{V}_{\prod_n p(x_n, y_n)}\left[\frac{1}{N}\sum_{n=1}^{N}\ell(f(\mathbf{x}_n;\boldsymbol{\theta}^*), y_n)\right] = \frac{1}{N}\mathbb{V}_{p(x,y)}[\ell(f(\mathbf{x};\boldsymbol{\theta}^*), y)] \qquad (3)$$

(Make sure you know how to prove this with all steps!)

- Want our estimator to always be as close to the true expected loss as possible.
- Small variance (spread!) $\implies$ large N.
- E.g. Chebyshev's inequality proves that estimate will be good with high probability.

Validation set size



Tension for small datasets:

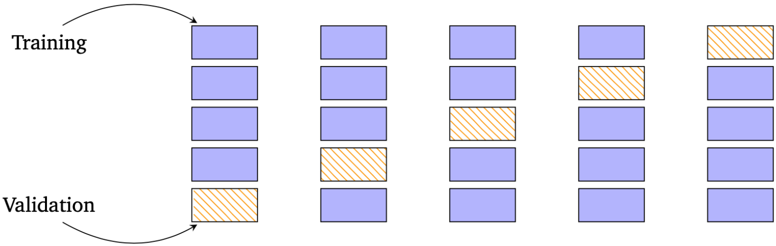
- Small validation set, large variance, may choose wrong model
- Large validation set size, small training set, may choose wrong parameters

Cross-validation

Can we somehow train on the whole dataset
and
validate on the of the whole dataset?

Cross-validation

Attempt to get more accurate estimate of validation/test loss, without reducing training set size too much. Cost: some small bias.



- Split data into train/validation sets in **multiple ways**
- Compute validation performance for each split
- Average to get **cross-validation** loss
- *Almost* like having K independent test sets, which would multiply the variance by $\frac{1}{K}$

Cross-validation

Procedure:

- Split data into train/validation in K different ways
- For each model
  - For each split
    - Find parameters of model
    - Compute loss on validation set
  - Calculate average validation loss for all splits (cross-validation loss)
- Pick model with lowest cross-validation loss

You can nest this as well, to get a cross-validation estimate of the test loss. Create an extra outer loop that splits data into train / test in K_{outer} different ways.

When to use cross-validation

- CV is expensive! It requires training a model K times.
- CV gives **biased** estimates!
  - But bias is generally small, so it is a reliable estimate.
  - But, difficult to prove things about!
- CV often gives smaller variance than a separate hold-out set of the same size.

So, rule of thumb:

If your dataset is small, it may be better to use crossvalidation for selecting hyperparameters and/or estimation of test error.