Computational
Social Science

Model Assessment & Selection

Roberto Cerina

08.02.2024

University of Amsterdam

Model Assessment & Selection

- ✏ *Selection*:
  estimate performance of a series of candidate models, with the goal of choosing the best one;

- ✏ *Assessment*:
  estimate the *generalisation error* of your chosen model.

Measures of Performance

- ✏ point-estimate for predicted value: $\hat{y}_i$
- ✏ $(1-\alpha)\%$ prediction interval for a given subject i: $(Q_\alpha(\hat{y}_i), Q_{1-\alpha}(\hat{y}_i))$

- ☞ estimated error (a.k.a. bias on a single data point) :
  $\hat{e}_i = \hat{y}_i - y_i$

- average direction of the error:
- ☞ Bias $= \frac{1}{n}\sum_i^n \hat{e}_i = \bar{\hat{e}}$
  - If training sample is *randomly selected*, and *large enough* $\rightarrow$ bias is entirely due to poor modeling assumptions / choices / framework;
  - non-random training sample / low *statistical power* $\rightarrow$ Bias $\uparrow$.

Measures of Performance

- Magnitude of the average error
  - root mean squared error:
    $RMSE = \sqrt{\frac{1}{n}\sum_i^n \hat{e}_i^2}$
  - mean absolute error:
    $MAE = \frac{1}{n}\sum_i^n \mid \hat{e}_i \mid$

Measures of Performance

- Ability to order observations correctly:
  - Pearson correlation coefficient:
    $\rho = \frac{\text{Cov}(\hat{y}, y)}{\sqrt{\text{Var}(\hat{y})\text{Var}(y)}} = \frac{\sum_i^n (\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_i^n (\hat{y}_i - \bar{\hat{y}})^2 \sum_i^n (y_i - \bar{y})^2}}$

- Probability with which the true value is contained in the prediction interval:
  - $\text{Coverage}_{1-\alpha} = \frac{1}{n} \sum_i^n [Q_\alpha(\hat{y}_i) \leq y_i \ \wedge Q_{1-\alpha}(\hat{y}_i) > y_i]$

Training, Validation & Test

- if *n* is large enough:

Training	Validation	Test
50% Used to estimate the candidate models' parameters	25% For each candidate model, make predictions, get performance score	25% For the best validated model, make predictions, get generalisation error

Theoretical Decomposition of the Total Error

- Assume a model of the classic form, where a given random variable y has the following DGP:

  $y = f(\boldsymbol{x}) + \epsilon$

  $\mathsf{E}(\epsilon) = 0$

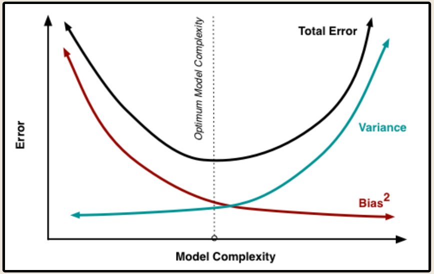
  $\mathsf{Var}(\epsilon) = \sigma^2$

- The expected squared error (Err) for a model generating prediction $\hat{f}(\boldsymbol{x})$, evaluated at a specific input point $\boldsymbol{x} = \boldsymbol{x}_0$, can be decomposed as follows:

$$
\begin{aligned}
\mathsf{Err} =& \mathsf{E}\left[\left(y_0 - \hat{f}(\boldsymbol{x}_0)\right)^2 \mid \boldsymbol{x} = \boldsymbol{x}_0\right] \\
=& \sigma^2 + \left(\mathsf{E}[\hat{f}(\boldsymbol{x}_0)] - f(\boldsymbol{x}_0)\right)^2 + \mathsf{E}\left(\hat{f}(\boldsymbol{x}_0) - \mathsf{E}[\hat{f}(\boldsymbol{x}_0)]\right)^2 \\
=& \sigma^2 + \mathsf{Bias}^2\left(\hat{f}(\boldsymbol{x}_0)\right) + \mathsf{Var}\left(\hat{f}(\boldsymbol{x}_0)\right) \\
=& \text{Irreducible Error} + \text{Bias}^2 + \text{Variance.}
\end{aligned}
$$

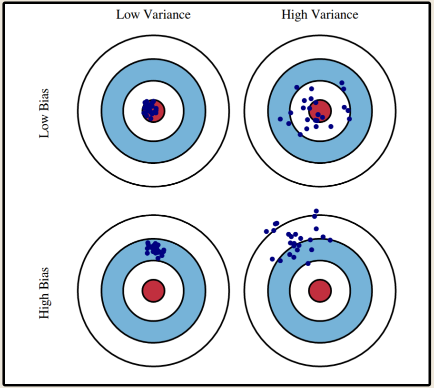
- Note: $\mathsf{E}[*]$ is the 'expectation' operator. It returns the expected value of the given random variable. Here we are taking the expected value across several hypothetical training sets.

Trade-off in Bias and Variance of Predictions



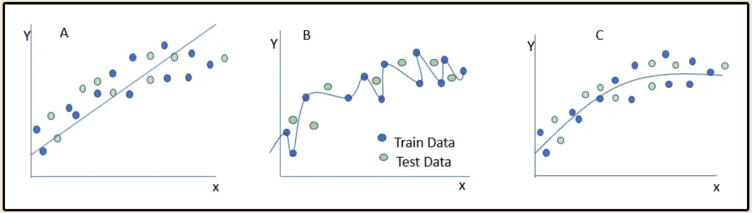
[0] https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

Trade-off in Bias and Variance of Predictions



[0] https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

Trade-off in Bias and Variance of Predictions



[0] https://medium.com/swlh/the-bias-variance-tradeoff-f24253c0ab45

Detecting & solving Under- / Over-fitting [1]

- High bias per prediction (High RMSE) - Figure A ← under-fitting ;
  - Symptoms:
    - Training error is higher than irreducible error.
  - ✔ Remedies:
    1. Use more complex model (e.g. kernelize, use non-linear models) ;
    2. Add features ;
    3. *Boosting*.
- High variance across predictions - Figure B ← over-fitting ;
  - Symptoms:
    - Training error is much lower than test error ;
    - Training error is lower than irreducible error ;
    - Test error is above irreducible error.
  - ✔ Remedies:
    1. Increase size of training data ;
    2. Reduce model complexity ;
    3. *Regularise* model coefficients.

[1] `https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html`

‘Optimism’ in the Training Sample

- Problem: training error $\bar{\text{err}}$ (in-sample) tends to under-estimates true generalisation error Err.

- e.g. $\bar{\text{err}} = \text{RMSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{n}\sum_i^n (\hat{y}_i - y_i)^2$

- In a given training set, we observe $\mathcal{T}^0 = \left(\boldsymbol{y} = \boldsymbol{y}^0, \boldsymbol{X} = \boldsymbol{X}^0\right)$ ...
- ...but due to the **irreducible error**, we could have as easily observed, for the same design matrix, $\mathcal{T}^k = \left(\boldsymbol{y} = \boldsymbol{y}^k, \boldsymbol{X} = \boldsymbol{X}^0\right)$, where $\boldsymbol{y}^k \neq \boldsymbol{y}^0$.

- A model trained on $\mathcal{T}^0$, for the same input $\boldsymbol{X} = \boldsymbol{X}^0$, we would tend to have a worse prediction for $\boldsymbol{y} = \boldsymbol{y}^k$ than we did for $\boldsymbol{y} = \boldsymbol{y}^0$:
- $\text{RMSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}^k) = \text{RMSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}^0) + \boldsymbol{\omega}^k$;
- $\boldsymbol{\omega}^k$ is the *optimism* associated with a specific ‘counterfactual value’ $\boldsymbol{y}^k$;
- Across all possible potential values of the outcomes $\boldsymbol{y}_k \in \{\boldsymbol{y}^1, ..., \boldsymbol{y}^K\}$, holding $\boldsymbol{X}$ fixed at $\boldsymbol{X} = \boldsymbol{X}^0$, the average optimism is:
  $\bar{\boldsymbol{\omega}} = \frac{1}{K}\sum_k^K \boldsymbol{\omega}^k$;

‘Optimism’ in the Training Sample

- It turns out that, for a random subject i:
  $\bar{\omega}_i = \frac{2}{n} \sum_i^n \text{Cov}(\hat{y}_i, y_i)$
- $\therefore$ the amount of training-sample ‘optimisim’ in the generalisation error depends on how much y_i influences $\hat{y}_i$ – that is the degree of *overfitting*.

Estimating Generalisation Error: Information Criteria

✏ For additive / linear models, we can estimate expected optimism:
$\hat{\bar{\omega}}_i = \frac{2}{n}\hat{d}\hat{\sigma}^2$

- $\hat{\sigma}^2$ is our best estimate of the irreducible error / population variance;
- $\hat{d}$ is a measure of model complexity, which represents the *effective number of parameters* used to fit the model;
- $\hat{d} = \frac{\sum_i^n \mathrm{Cov}(\hat{y}_i, y_i)}{\sigma^2}$

Comparing Models in-Sample: Information Criteria

- We can use a 'corrected' version of our in-sample error estimate, which accounts for model-complexity, to discriminate across candidate models and avoid over-fitting:

  $C = \bar{\text{err}} + \frac{2}{n}\hat{d}\hat{\sigma}^2$

- Compared to the 'uncorrected' form, this will favour a more 'parsimonious' model, for the same level of error – as it stands to reason its 'optimisim' bias is lower.

- Example: *Akaike Information Criteria*

- AIC uses the *negative log-likelihood* of the data under the model as the metric for error:

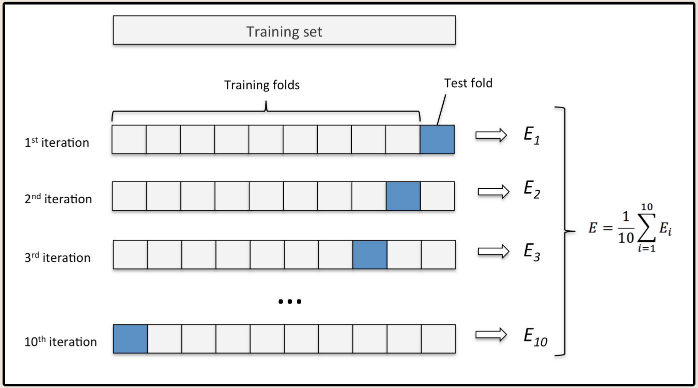
  $\text{AIC} = -\frac{2}{n}log(\mathcal{L}_i) + \frac{2}{n}\hat{d}$

- lower AIC = better fit – models with high complexity are penalised and have larger AIC values.

- Nnote: **do not** compare AIC from models with different likelihood (i.e. Bernoulli v. Gaussian) or with different training sets – they are on different scales...

Estimating Generalisation Error: K-fold Cross-Validation

- ❢ n is typically not large enough , and signal-to-noise ratio not strong enough, to have large-enough training, validation and test datasets...
- ✔ K-fold cross-validation allows to estimate generalisation error more 'economically'...



[1] http://karlrosaen.com/ml/learning-log/2016-06-20/

Estimating Generalisation Error: K-fold Cross-Validation

- $K = N$: 'Leave-one-out' CV !
- generates unbiased estimate of generalisation error for 'large enough' n ...
- ... but can have large variance if model is highly complex ...
- ... and can be extremely computationally taxing...

Estimating Generalisation Error: K-fold Cross-Validation

✔ $K = 5$ or $K = 10$ tend to be good compromises. [2]

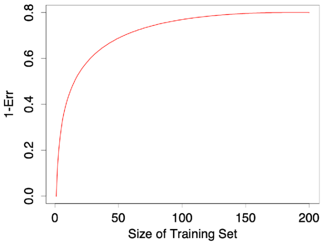


FIGURE 7.8. *Hypothetical learning curve for a classifier on a given task: a plot of* $1 - \mathrm{Err}$ *versus the size of the training set* N*. With a dataset of* 200 *observations,* 5*-fold cross-validation would use training sets of size* 160*, which would behave much like the full set. However, with a dataset of* 50 *observations fivefold cross-validation would use training sets of size* 40*, and this would result in a considerable overestimate of prediction error.*

[2] Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).