Model Selection

Hamid Dehghani
School of Computer Science
Birmingham
September 2020

Slides adapted from Iain Styles, School of Computer Science

UNIVERSITY OF BIRMINGHAM

Intended Learning Outcome

- Understand and be able interpret and apply the principles of empirical model selection

UNIVERSITY OF BIRMINGHAM

Model Selection & Evaluation

- We try to find function *f(x)* to match underlying data generating function *h(x)*
- If we know *h* we can easily choose *f*
- But normally we do not know *h*
  - *f* then has to be determined experimentally
- <u>Validation and Testing</u>
- Checking the model's ability to predict unseen data

Approach 1: Train-Validate-Test

- Common for very large data sets
  - Often used in machine learning competitions
- Some of the data used to train the model
- Some of the data used to evaluate whether that model is any good
- Some of the data used to test what we think is the best model

Approach 1: Train-Validate-Test

- Partition a dataset D into:
  - A training set T, randomly sampled from D
  - A validation set V ,randomly sampled from D
  - A test, or evaluation set $E = D - T - V$
- Define a set of models $\{M_i\}_{k=1}^{K}$
- and a loss function L (least-square for regression

UNIVERSITY OF BIRMINGHAM

Approach 1: Train-Validate-Test

- Training set T used to optimise model parameters
- Validation set V used to select optimal model
  - hyperparameter optimisation
- Select the choice of hyperparameters that best allows the model learned on T to generalise to V
- Evaluate on E to assess how well the model performs on unseen data
- Ultimate test of its ability to generalise
- Guards against overfitting of the hyperparameters to V

Approach 1: Train-Validate-Test

- *T* must be big enough to fully represent the data:
  - an 80-10-10 split is quite common
- *D* must be randomised to ensure *T* , *V*, *E* represent the data
  - For example: sample from across the data's domain x and range y
- Example:
  - Twenty data points and split them into a training set *T* of ten points, a validation set *V* of five points, and a test set *E* of 5 points.

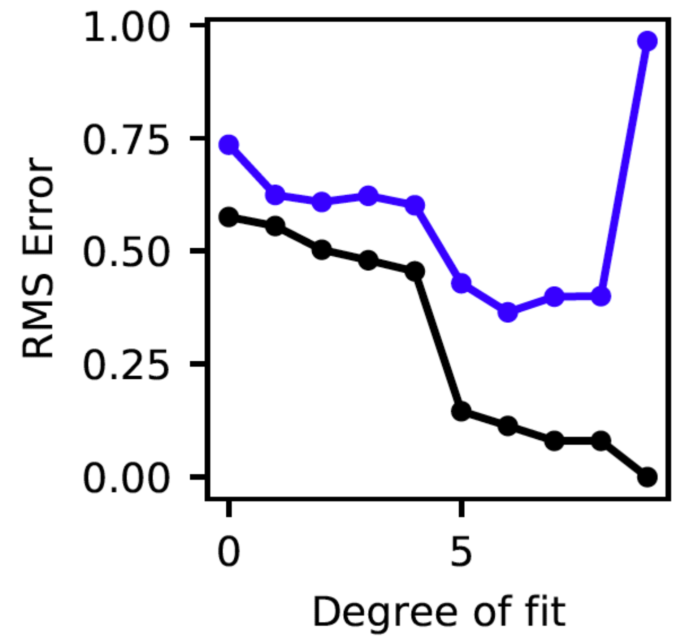
Approach 1: Train-Validate-Test

Data: Set of models $\{\mathcal{M}_i\}_i$
Data: Dataset $\mathcal{D}$ split into training ($\mathcal{T}$), validation ($\mathcal{V}$), and test/evaluation ($\mathcal{E}$) sets.
Result: Identification of model $\mathcal{M}*$ with best predictive power.
for *each model* $\mathcal{M}_i$ **do**
    Train $\mathcal{M}_i$ on $\mathcal{T}$;
    Compute model loss $\mathcal{L}_{\mathcal{T}}$ on training set $\mathcal{T}$;
    Compute model loss $\mathcal{L}_{\mathcal{V}}$ on evaluation set $\mathcal{V}$;
end
Select model $\mathcal{M}*$ with best overall performance on training and validation sets.;
Compute loss on test set $\mathcal{E}$ to determine final model performance;

UNIVERSITY OF BIRMINGHAM

Approach 1: Train-Validate-Test

- **Validation**
  - Training error continues to improve
  - Validation error also improves but then gets dramatically worse
  - Model over-fits the training data
  - Models trend + noise so cannot generalise
  - **Occam's razor**: choose simplest model that performs well: M = 5?



UNIVERSITY OF BIRMINGHAM

Approach 1: Train-Validate-Test

- Notes to take-away
  - Validation set is used to select the best model
  - Test set is used right at the end to ensure that the model generalises beyond the validation set
  - Avoids hyperparameter over-fitting

- What if we had even less data?

Approach 2: Cross-Validation

- **Small data**
  - – may not be able to adequately split into representative groups
- In cross-validation we split the data into *K* folds.
- Train models on K - 1 of the folds
- Validate on the remaining fold
- Use each of the folds in turn as the validation set
- Select the model that gives the best average performance

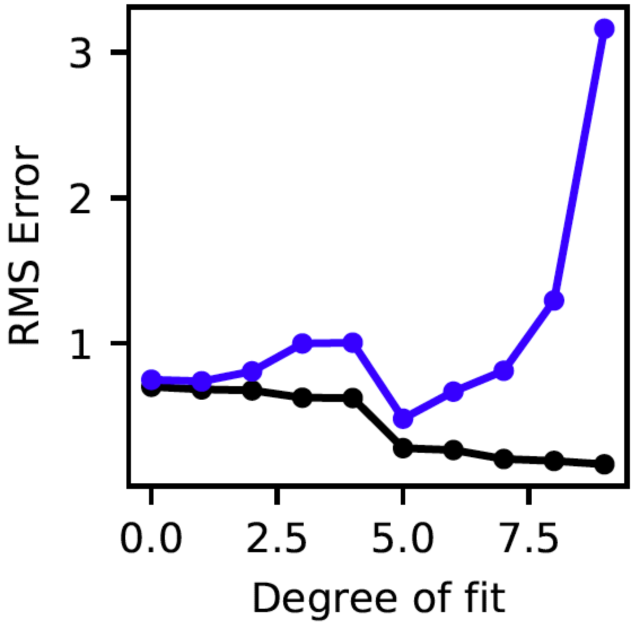
Approach 2: Cross-Validation

Data: Set of models $\{\mathcal{M}_i\}_i$
Data: Dataset $\mathcal{D}$ split into cross-validation ($\mathcal{V}$), and test/evaluation ($\mathcal{E}$) sets.
Data: Number of folds, K
Result: Identification of model $\mathcal{M}*$ with best predictive power.
Divide $\mathcal{C}$ into K folds $\{c_k\}_{k=1}^K$ such that $\mathcal{C} = \bigcup_{k=1}^K c_k$;
for *each model* $\mathcal{M}_i$ **do**
  for $k = 1 \rightarrow K$ **do**
    Train $\mathcal{M}_i$ on training set $\mathcal{C} - c_k$;
    Compute model loss $\mathcal{L}_\mathcal{T}$ on training set $\mathcal{C} - c_k$;
    Compute model loss $\mathcal{L}_\mathcal{V}$ on evaluation fold c_k;
  end
end
Select model $\mathcal{M}*$ with best overall performance on training and validation sets;
Compute loss on test set $\mathcal{E}$ to determine final model performance;

UNIVERSITY OF BIRMINGHAM

Approach 2: Cross-Validation

- **Validation**
- K=5-fold cross validation on 20-pt dataset
- Training set is larger 16 points in each round
- Results are averaged over the folds
- Validation error also improves but then gets dramatically worse
- Similar conclusions can be drawn
  - models of order 5 perform well on both the training and validation sets



UNIVERSITY OF BIRMINGHAM

Summary

- Splitting the data up allows us to experimentally determine the optimal model
- Can be computationally expensive - especially cross validation
- Next Lecture
  - controlling models with prior knowledge
  - Regularisation, leading into a Bayesian approach to regression

UNIVERSITY OF BIRMINGHAM