Model selection

Nathanaël Carraz Rakotonirina

Mathématiques Informatique et Statistique Appliquées (MISA)
Université d'Antananarivo

Generalization

Goal

We want to perform well on new previously unseen inputs (not just those on which our model was trained.). This is called **generalization**.

- The **training error** is the error measure on the training set.
- The **generalization error** or **test error** is the expected value of the error on a new input (expectation is taken across different possible inputs, drawn from the distribution of inputs we expect the system to encounter in practice.).

  We typically estimate the generalization error of a machine learning model by measuring its performance on a **test set**.

Bias/variance trade-off (underfitting/overfitting)

A model performs well if:

- the training error is small
- the gap between training and test error is small

Underfitting

- The training error is large.
- The model has large **bias**.
- It cannot capture the structures of the data (model is too "simple").

Overfitting

- The gap between the training error and test error is large.
- The model has large **variance**.
- It captures the structures that happened to be present in the small, finite training set, but that do not reflect the wider structures (model is too "complex").

Regularization

One way to solve the overfitting problem and control complexity of the model is to use **regularization**.

Regularization

Regularization is any modification we make to a machine learning system to reduce its generalization error but not its training error.

Common regularization approaches:

- adding a penalty term (like a norm penalty) to the cost function
- dropout for neural networks

No Free Lunch Theorem

Theorem

Averaged over all possible data generating distributions, every classification model has the same error rate when classifying previously unobserved points.

This means:

- No machine learning model is universally any better than any other.
- The choice of the best model depends on the specific task and the nature of the data.

Model selection

Goal

Given a set of models $\{\mathcal{F}_1, \mathcal{F}_2, ...\}$, we want to select the model giving best performance on unseen data.

Example :

- Decide between using different forms of models (an SVM, a neural network or logistic regression)
- Choosing the best hyperparameter for a specific model form (choosing λ for ridge regression, choosing C and the kernel for SVM)

Validation set

We cannot choose the best model using the test set (overfit). Instead, we do so on a hold-out **validation set**.

1. Randomly split the data into the training set $\mathcal{D}_{train}$, the validation set $\mathcal{D}_{val}$ and the training set $\mathcal{D}_{test}$ ($\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{val} \cup \mathcal{D}_{test}$)
2. Train each model on $\mathcal{D}_{train}$
3. Evaluate the performance on $\mathcal{D}_{val}$
4. Select the model with the best performance on $\mathcal{D}_{val}$
5. Test the selected model (only when satisfied) on $\mathcal{D}_{test}$

Warning !

Use the test set $\mathcal{D}_{test}$ only once.

Cross-validation

When the data is small, we might consider **k-fold cross validation**.

1. Randomly split the data into the training and test sets ($\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$)
2. Randomly split $\mathcal{D}_{train}$ into k disjoint subsets ($\mathcal{D}_{train} = \mathcal{D}_1 \cup ... \cup \mathcal{D}_k$)
3. Evaluate each model as follows
   - For $i = 1$ to k
     - Put aside $\mathcal{D}_i$
     - Train the model on the $k - 1$ remaining subsets
     - Evaluate the performance on $\mathcal{D}_i$
   - Average the k performance measures
4. Select the model with the best performance
5. Test the selected model (only when satisfied) on $\mathcal{D}_{test}$

Further works

Explore

- Statistical learning theory (bound on generalization error)
- Error analysis
- Practical machine learning projects
- Automated machine learning (AutoML)