Cross validation

The model selection problem

Objective

- Often necessary to consider many different models (e.g., types of classifiers) for a given problem.
- Sometimes "model" simply means particular setting of **hyper-parameters** (e.g., k in k-NN, number of nodes in decision tree).

Terminology

The problem of choosing a good model is called **model selection**.

Model selection by hold-out validation

(Henceforth, use h to denote particular setting of hyper-parameters / model choice.)

Hold-out validation

Model selection:

1. Randomly split data into three sets: training, validation, and test data.

Training	Validation	Test

2. Train classifier $\hat{f}_h$ on Training data for different values of h.
3. Compute Validation ("hold-out") error rate for each $\hat{f}_h$: $\mathrm{err}(\hat{f}_h, \text{Validation})$.
4. Selection: $\hat{h}$ = value of h with lowest Validation error rate.
5. Train classifier $\hat{f}$ using $\hat{h}$ with Training and Validation data.

Model assessment:

6. Finally: estimate true error rate of $\hat{f}$ using test data.

Main idea behind hold-out validation

Training	Validation	Test

Classifier $\hat{f}_h$ trained on Training data $\longrightarrow \mathrm{err}(\hat{f}_h, \text{Validation})$.

Training and Validation	Test

Classifier $\hat{f}_h$ trained on Training and Validation data $\longrightarrow \mathrm{err}(\hat{f}_h, \text{Test})$.

The hope is that these quantities are similar!

(Making this rigorous is actually rather tricky.)

Beyond simple hold-out validation

Standard hold-out validation:

Training	Validation	Test

Classifier $\hat{f}_h$ trained on Training data $\longrightarrow \mathrm{err}(\hat{f}_h, \text{Validation})$.

Could also swap roles of Validation and Training:

- train $\hat{f}_h$ using Validation data, and
- evaluate $\hat{f}_h$ using Training data.

Training	Validation	Test

Classifier $\hat{f}_h$ trained on Validation data $\longrightarrow \mathrm{err}(\hat{f}_h, \text{Training})$.

Idea: *Do both*, and average results as overall validation error rate for h.

Model selection by K-fold cross validation

Model selection:

1. Set aside some test data.
2. Of remaining data, split into K parts ("folds") $S_1, S_2, \dots, S_K$.
3. For each value of h:
   - For each $k \in \{1, 2, \dots, K\}$:
     - Train classifier $\hat{f}_{h,k}$ using all S_i except S_k.
     - Evaluate classifier $\hat{f}_{h,k}$ using S_k: $\mathrm{err}(\hat{f}_{h,k}, S_k)$

     Example: $K = 5$ and $k = 4$

Training	Training	Training	Validation	Training

   - K-fold cross-validation error rate for h: $\frac{1}{K}\sum_{k=1}^{K} \mathrm{err}(\hat{f}_{h,k}, S_k)$.
4. Set $\hat{h}$ to the value h with lowest K-fold cross-validation error rate.
5. Train classifier $\hat{f}$ using selected $\hat{h}$ with all $S_1, S_2, \dots, S_K$.

Model assessment:

6. Finally: estimate true error rate of $\hat{f}$ using test data.

How to choose K?

Argument for small K

Better simulates "variation" between different training samples drawn from underlying distribution.

$K = 2$

Training	Validation
Validation	Training

$K = 4$

Validation	Training	Training	Training
Training	Validation	Training	Training
Training	Training	Validation	Training
Training	Training	Training	Validation

Argument for large K

Some learning algorithms exhibit *phase transition* behavior
(e.g., output is complete rubbish until sample size sufficiently large).
Using large K best simulates training on all data (except test, of course).

In practice: usually $K = 5$ or $K = 10$.

Some pitfalls

Pitfalls:

- considering *too many* different models can lead to overfitting, even with cross-validation.
- adaptively choosing which models to consider *after having looked at the validation/test data* breaks independence assumptions!

Key takeaways

1. Model selection problem.
2. Procedures for and intuitions behind hold-out and K-fold cross validation.
3. High-level idea of limitations of hold-out and cross validation.