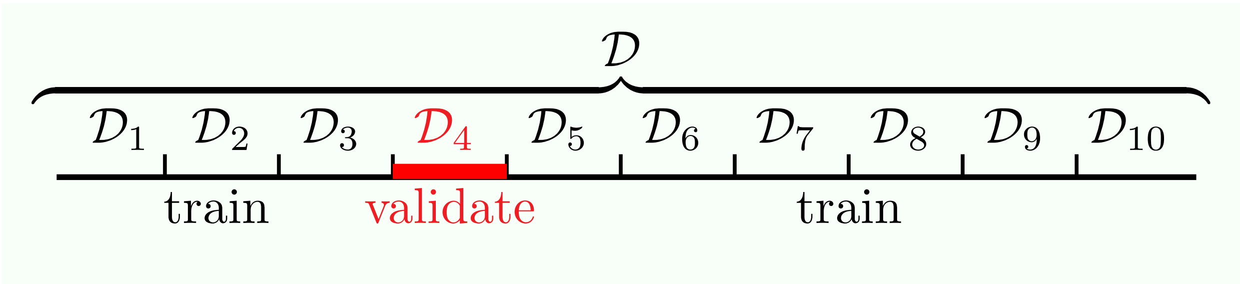
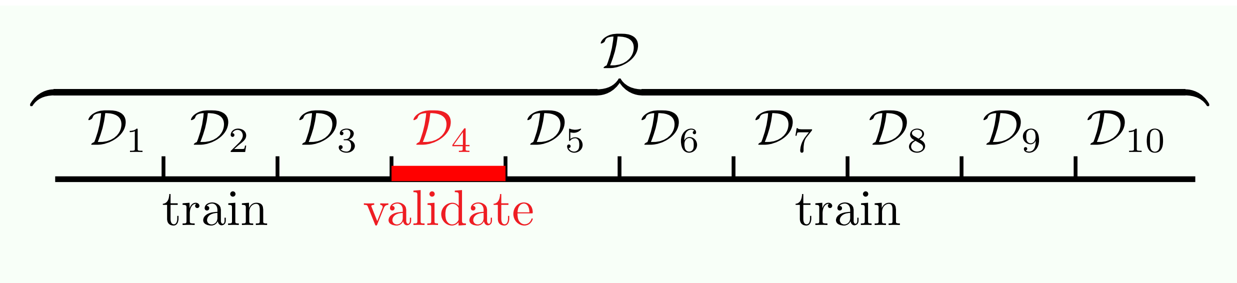
Evaluating and using ML classifiers: model selection using cross validation and data snooping



Reminder: Cross validation

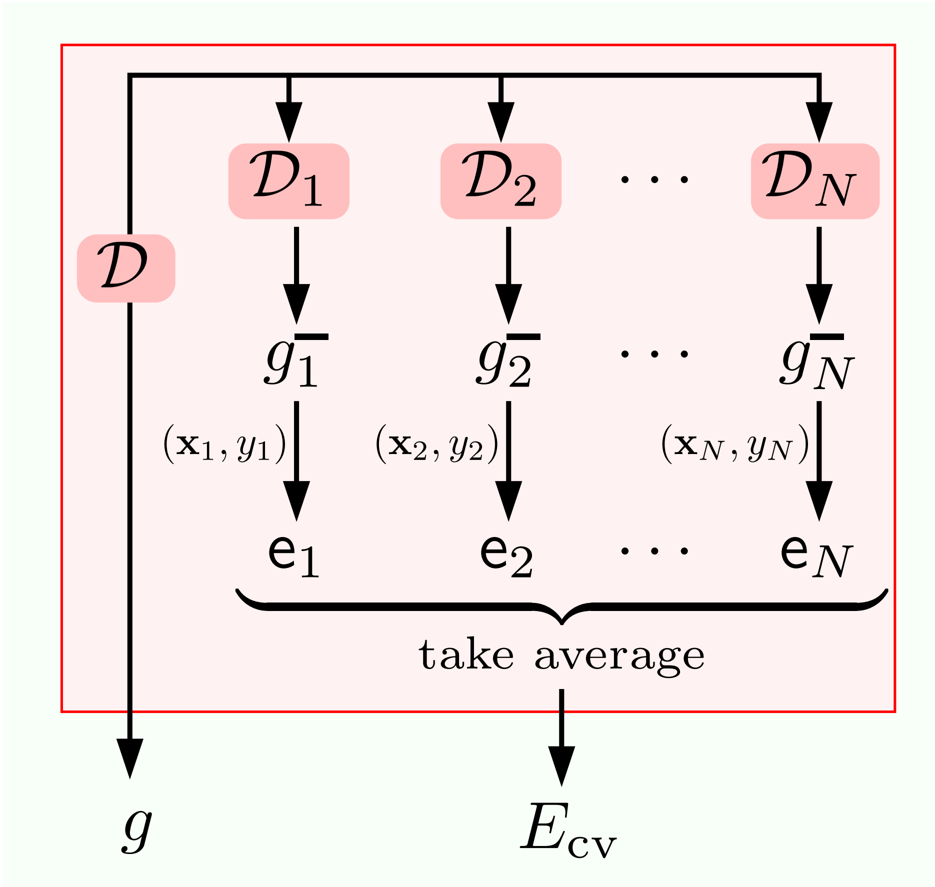
Cross validation:

- Randomly partition the data into k parts ("folds").
- Set one fold aside for evaluation and train a model on the remaining k-1 folds and evaluate it on the held-out fold.
- Repeat until each fold has been used for evaluation



Stratified-cross validation aims at achieving roughly the same class distribution in each fold.

Model selection using cross-validation



Now we have to ask ourselves how well do we expect that classifier to perform. But E_{cv} is a biased estimate of performance of the classifier trained using the chosen parameters.

Model selection

The task:

For each classifier compare the accuracy of the best parameter setting (estimated using cross-validation or a test set)

So, assuming we are comparing two classifiers, this means we are making the following comparison:

$$\max(s_1,\ldots,s_m) \quad \text{vs} \quad \max(t_1,\ldots,t_m)$$

In computing the maximum we are using information about the labels!

Cross validation tells you how well the classifier is performing on a given setting of classifier parameters.

Model selection

How to select and evaluate machine learning models:

1. Set aside data for training / validation / testing
2. For each value of classifier hyperparameters: train a model on the training set and evaluate it on the validation set.
3. Choose the best performing model
4. Use its parameters to train a model on the training + validation set and evaluate it on the test set.

5. Train a model on all the data and deliver it to your customer along with the estimate of its accuracy.

Model selection

How to select and evaluate machine learning models:

1. Set aside data for training / validation / testing
2. For each value of classifier hyperparameters: train a model on the training set and evaluate it on the validation set.
3. Choose the best performing model
4. Use its parameters to train a model on the training + validation set and evaluate it on the test set.

5. Train a model on all the data and deliver it to your customer along with the estimate of its accuracy.

Potential issue: may not have sufficient data for having separate training / validation / test sets

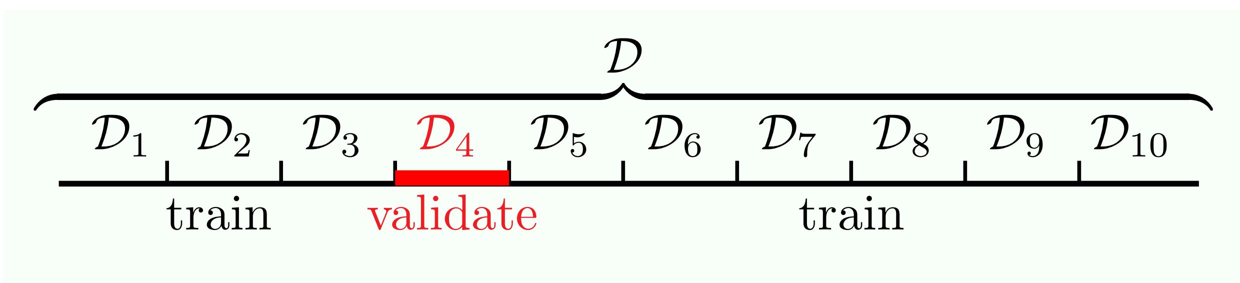
Two ways of doing cross validation

External cross validation:

- Perform cross validation across various settings of classifier parameters and report the best result you got

Internal cross validation (nested CV):

- For each fold, perform cross-validation on the training data, and train a classifier on the best set of parameters for that fold
- This evaluates the training procedure



Internal vs External cross-validation estimates

Data Set	External	Internal	Bias
banana	10.355 ± 0.146	10.495 ± 0.158	0.140 ± 0.035
breast cancer	26.280 ± 0.232	27.470 ± 0.250	1.190 ± 0.135
diabetis	22.891 ± 0.127	23.056 ± 0.134	0.165 ± 0.050
flare solar	34.518 ± 0.172	34.707 ± 0.179	0.189 ± 0.051
german	23.999 ± 0.117	24.217 ± 0.125	0.219 ± 0.045
heart	16.335 ± 0.214	16.571 ± 0.220	0.235 ± 0.073
image	3.081 ± 0.102	3.173 ± 0.112	0.092 ± 0.035
ringnorm	1.567 ± 0.058	1.607 ± 0.057	0.040 ± 0.014
splice	10.930 ± 0.219	11.170 ± 0.280	0.240 ± 0.152
thyroid	3.743 ± 0.137	4.279 ± 0.152	0.536 ± 0.073
titanic	22.167 ± 0.434	22.487 ± 0.442	0.320 ± 0.077
twonorm	2.480 ± 0.067	2.502 ± 0.070	0.022 ± 0.021
waveform	9.613 ± 0.168	9.815 ± 0.183	0.203 ± 0.064

Table 8: Error rate estimates for kernel ridge regression over thirteen benchmark data sets, for model selection schemes that are internal and external to the cross-validation process. The results for each approach and the relative bias are presented in the form of the mean error rate over for 100 realisations of each data set (20 in the case of the image and splice data sets), along with the associated standard error.

Table from
On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation
Gavin C. Cawley, Nicola L.C. Talbot, JMLR 11:2079–2107, 2010.
http://jmlr.org/papers/v11/cawley10a.html

Internal cross-validation

Notice that each train/test fold may get different parameter settings.

That's fine!

This results in a "parameterless" algorithm that internally sets parameters for each data set it gets

What to do for the system you are deploying

Use external cross-validation to determine good parameters

Train your model on ALL the data.

Provide your customer with the results of internal-cross validation as estimates of future performance.

Data snooping

As machine learning researchers we often do the following:

1. Do a proper internal cross-validation experiment
2. Improve the algorithm/features; goto 1

Is there an issue?

Data snooping

As machine learning researchers we often do the following:

1. Do a proper internal cross-validation experiment
2. Improve the algorithm/features; goto 1

Is there an issue?

(Machine Learning's dirty secret!)

Correct classifier evaluation

When running experiments consider the following question:

On each fold of cross-validation, did I ever access in any way the label of a test case?

Any preprocessing done over the entire data set (feature selection, parameter tuning, threshold selection) must not use labels

Using repository data for classifier evaluation

Pros:

- Very easy to implement
- Data from real applications
- Facilitates replication and comparison of results

Cons:

- Community experiment/multiplicity effect: since so many experiments are run on the same data set, by chance, some will yield interesting (though meaningless) results

Model selection in scikit-learn

Is nested cross-validation difficult in scikit-learn?

NO!