One solution: “k-fold cross validation”

- Used to better estimate generalization accuracy of model
- Used to learn hyper-parameters of model (“model selection”)

Using k-fold cross validation to estimate accuracy

- Each example is used both as a training instance and as a test instance.
- Instead of splitting data into “training set” and “test set”, split data into k disjoint parts: S1, S2, ..., Sk.
- For i = 1 to k

    Select Si to be the “test set”. Train on the remaining data, test on Si, to obtain accuracy Ai .
- Report 1/k*∑Ai as the final accuracy.

Using k-fold cross validation to learn hyper-parameters
(e.g., learning rate, number of hidden units, SVM kernel, etc.)

- Split data into training and test sets. Put test set aside.
- Split training data into k disjoint parts: S1, S2, ..., Sk.
- Assume you are learning one hyper-parameter. Choose R possible values for this hyper parameter.
- For j = 1 to R

```
    For i = 1 to k
       Select Si to be the “validation set”
        Train the classifier on the remaining data using the jth value of the hyper parameter
        Test the classifier on Si, to obtain accuracy Ai ,j.
    Compute the average of the accuracies: Aj = 1/k*∑Ai, j
Choose the value j of the hyper-parameter with highest Aj.
Retrain the model with all the training data, using this value of the hyper-parameter.
Test resulting model on the test set.
```

precision: when you want to be sure saying correct is correct.
recall: ok with getting incorrect, because you get all the correct.

```
TP | FN
---+---
FP | TN
```

A = (TP + TN) / (total)
P = TP / (TP + FP)
R = TP / (TP + FN)

BIAS: Classifier is not powerful enough to represent the true function; that is, it under fits the function

    trained using linear instead of more complicated kernel, C parameter set too low

Variance: Classifier’s hypothesis depends on specific training set; that is, it over fits the function

    training set small, test set is small

Noise: Underlying process generating data is stochastic, or data has errors or outliers

    values in training set came from imprecise measurement, class labels entered incorrectly

ROC
TPR = TP / (TP+FN) = y
FRP = FP / (TP+FN) = x