Model Selection and Cross-Validation

Mikio L. Braun and Klaus-Robert Müller

Choosing Free Parameters

Many learning algorithms have “free parameters”, for example

- Regularization constants
- Selection of basis functions / kernels
- preprocessing

Choosing such parameters based on the training error alone will usually not work, because more complex models will invariably lead to smaller training error.

For example, if we fit with polynomials of degree d, the fit error will decrease monotonically.

(plot)

Therefore, we need a reliable measure, to estimate the expected risk

In the following:

- cross-validation
- correction terms

Overfitting from a Statistical Perspective

The main problem is, abstractly speaking, that in general

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} h(X_i) < E(h(X)).$$

Based on the specific realization, the inequality does not have to hold, but

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} h(X_i) \leq \frac{1}{n} \sum_{i=1}^{n} h(X_i)$$

for each $h \in \mathcal{H}$, and therefore

$$\lim_{n \to \infty} \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} h(X_i) \leq E(h(X)).$$

An extreme example

Consider identical dice.

Assume you suspect that the dice are loaded and you want to select the one giving the smallest number of pits.

You try each dice once and take the one with the smallest result. But all dice are equal.

Test Sets

The expected risk can be estimated reliably on an *independent sample*:

You train on $X_1, Y_1, \ldots, X_n, Y_n$ and get f, and then compute

$$\hat{\mathcal{R}}'(f) = \frac{1}{m} \sum_{i=1}^{m} L(f(X_i'), Y_i').$$

Where $X_1', Y_1', \ldots, X_m', Y_m'$ are i.i.d. samples from the same distribution as the X_i, Y_i.

You can always obtain an independent sample by splitting your data set!

k-Fold Cross-Validation

k-fold cross-validation (cv) is a method to estimate suitable values for free parameters of a learning method.

It is in general quite computationally intensive and does not work well for small data sets (up to a few hundred points).

k-Fold Cross-Validation

Split a data set k times (for example $k = 5$):

$$\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline\end{array}$$

For a fixed choice of parameters, iterate k times:
Train on $k - 1$ parts, and evaluate on the remaining part:

Iteration 1: train on $\begin{array}{|c|c|c|c|}\hline 2 & 3 & 4 & 5 \\ \hline\end{array}$, test on $\begin{array}{|c|}\hline 1 \\ \hline\end{array}$

Iteration 2: train on $\begin{array}{|c|c|c|c|}\hline 1 & 3 & 4 & 5 \\ \hline\end{array}$, test on $\begin{array}{|c|}\hline 2 \\ \hline\end{array}$

Iteration 3: train on $\begin{array}{|c|c|c|c|}\hline 1 & 2 & 4 & 5 \\ \hline\end{array}$, test on $\begin{array}{|c|}\hline 3 \\ \hline\end{array}$

Iteration 4: train on $\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 5 \\ \hline\end{array}$, test on $\begin{array}{|c|}\hline 4 \\ \hline\end{array}$

Iteration 5: train on $\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline\end{array}$, test on $\begin{array}{|c|}\hline 5 \\ \hline\end{array}$

k-Fold Cross-Validation

You obtain k estimates $\hat{\mathcal{R}}_1, \ldots, \hat{\mathcal{R}}_k$ for each possible parameter choice.

Repeat the procedure for a number of choices and choose the one with the smallest estimated risk (on average, median, etc.)

Issues with cross-validation

Note that the CV test error is again too optimistic.

As a general rule:

$$\boxed{\text{Risk estimates are only reliable if no selection based on the estimates have been performed!}}$$

$\rightsquigarrow$ For assessment of the selected models, we need a further *validation set*.

Issues with CV: computational demands

- Note that for each parameter candidate, you have to train and evaluate k times.
- CV scales exponentially with the number of free parameters if you perform simple grid search.
- For some algorithms, it is possible to compute *leave-one-out cross-validation* (i.e. $k = n$) more efficiently.

Issues with CV: choice of k

- For small k, the training set is much smaller than the original data set. Therefore, the test error is *biased* (meaning that it's expectation is not the real test error)
- For large k, the bias becomes small (for $k = n$, the estimator is *asymptotically unbiased*), but the test error has a high variance, because the error is estimated from only a few points.
- Typically, on chooses k somewhere between 5 and 10.

A Final Word on Preprocessings

The rule that the risk estimates are only reliable if no selection is performed based on the risk also extends to preprocessings:

> Risk estimates are only reliable if the information in the test data set has not been used in any way.

This in particular includes preprocessing such as denoising.

WRONG

- Denoise whole data set
- Split data set to perform cross-validation
- ...

CORRECT

- Split data set to perform cross-validation
- Denoise training set
- Learn classifier
- For prediction, denoise test set using the same parameters as on the training set
- Predict on denoised data
- ...

Correcting the Prediction Error

- We'll assume for now, that the $x_1, \ldots, x_n$ are fixed, and only the $Y_1, \ldots, Y_n$ have noise.
- We also consider the quadratic loss only for now.
- In this setting, the difference between training and test error can be analyzed much better.

We are interested in comparing the two quantities:

- The *expected test error*:

$$E(\hat{\mathcal{R}}'(\hat{f})) = E\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \hat{f}(x_i))^2\right]$$

  where the Y_i' is another independent sample of labels. Note that $\hat{f}$ depends on all the X_i, Y_i, but not on the Y_i'.

- The *expected training error*:

$$E(\hat{\mathcal{R}}(\hat{f})) = E\left[\frac{1}{n}\sum_{i=1}(Y_i - \hat{f}(x_i))^2\right]$$

Optimism of the Training Error

The *optimism op* is defined as

$$op = E(\hat{\mathcal{R}}'(\hat{f})) - E(\hat{\mathcal{R}}(\hat{f})).$$

It holds that

$$op = \frac{2}{n}\sum_{i=1}^{n} Cov(Y_i, \hat{f}(x_i)).$$

Proof

It holds that

$$op = \frac{1}{n}\sum_{i=1}^{n} E\left[(Y_i' - \hat{f}(x_i))^2 - (Y_i - \hat{f}(x_i))^2\right].$$

We first consider the term in the expectation:

$$(Y_i' - \hat{f}(x_i))^2 - (Y_i - \hat{f}(x_i))^2 =$$
$$Y_i'^2 - 2Y_i'\hat{f}(x_i) + \hat{f}^2(x_i) - Y_i^2 + 2Y_i\hat{f}(x_i) - \hat{f}^2(x_i),$$

and the $\hat{f}^2(x_i)$ terms cancel out.

Next, we take the expectation

$$E(Y_i'^2) - 2E(Y_i'\hat{f}(x_i)) - E(Y_i^2) + 2E(Y_i\hat{f}(x_i)).$$

Now, note that Y_i' and Y_i have the same distribution, therefore $E(Y_i'^2) = E(Y_i^2)$. Furthermore, $E(Y_i'\hat{f}(x_i)) = E(Y_i')E(\hat{f}(x_i))$, because f does not depend on the Y_i'.

$$= 2E(Y_i\hat{f}(x_i)) - 2E(Y_i)E(\hat{f}(x_i)) + E(Y_i^2) - E(Y_i^2)$$
$$= 2Cov(Y_i, \hat{f}(x_i)).$$

$\square$

In summary,

$$\text{expected test error} = \text{expected training error} + \frac{2}{n}\sum_{i=1}^{n} Cov(Y_i, \hat{f}(x_i)).$$

The stronger the dependency between $\hat{f}(x_i)$ and Y_i, the stronger the optimism of the training error.

Linear Learning Methods

For linear methods where the predictions on the training set depend linearly on the input labels, the covariance can be computed exactly.

For example, the least squares regression algorithm learns a function of the form

$$f(x) = \sum_{j=1}^{d} \alpha_j \phi_j(x),$$

where ϕ_j are appropriate basis functions (coordinates, polynomials, etc.).

Linear Learning Methods

For such methods, we have seen that the α which minimizes the quadratic loss can be computed in closed form as

$$\alpha = AY,$$

where $Y = (Y_1, \ldots, Y_n)$, and A is some matrix.

The *in-sample predictions* $\hat{Y} = (\hat{f}(x_1), \ldots, \hat{f}(x_n))$ are then given by

$$\hat{Y} = \Phi AY = SY \quad \text{``hat matrix''}$$

where $\Phi_{ij} = \phi_j(x_i)$, for $1 \leq i \leq n$, $1 \leq j \leq d$.

We assume that

$$f(x) = Y_i + \varepsilon_i,$$

where the ε_i are i.i.d. normally distributed with mean zero and variance σ_ε^2.

(This is a standard setup for regression).

If $\hat{Y} = SY$, then

$$\sum_{i=1}^{n} Cov(Y_i, \hat{Y}) = trace(S)\sigma_\varepsilon^2.$$

Proof

Note that $\sum_{i=1}^{n} Cov(Y_i, \hat{Y}_i) = trace\, Cov(Y, \hat{Y})$, and

$$Cov(Y, \hat{Y}) = E(Y\hat{Y}^t) - E(Y)E(\hat{Y})^t.$$

Let us define

$$F = (f(x_1), \ldots, f(x_n)),$$
$$\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$$

such that $Y = F + \varepsilon$, $\hat{Y} = S(F + \varepsilon)$.

Then:

$$\begin{aligned} E(Y\hat{Y}^t) &= E(YY^tS^t) = E((F+\varepsilon)(F+\varepsilon)^tS^t) \\ &= FF^tS^t + E(\varepsilon\varepsilon^t)S^t + E(\varepsilon)F^tS^t + FE(\varepsilon)^tS^t. \end{aligned}$$

Note that $E(\varepsilon) = 0$, and that $E(\varepsilon\varepsilon^t) = \sigma_\varepsilon^2 I$.

Now,

$$\begin{aligned} Cov(Y, \hat{Y}) &= E(Y\hat{Y}^t) - E(Y)E(\hat{Y})^t \\ &= FF^tS^t + \sigma_\varepsilon^2 S^t - FF^tS^t \quad = \sigma_\varepsilon^2 S^t. \end{aligned}$$

□

Comments

- For the unregularized case, $trace(S) = d$, the number of basis functions.
- For this reason, $trace(S)$ is also often called the *effective degrees of freedom*
- Also for non-linear methods, a similar quantity can be defined by

$$df = E\left[trace\left(\frac{\partial \hat{Y}}{\partial Y}\right)\right]$$

(again for fixed x_i).

Correcting the Optimism

of the training error for linear learning methods

The idea is to estimate *op* and then get an estimate of the true test error by adding it to the training error.

It holds that

$$E(\hat{\mathcal{R}}'(\hat{f})) = E(\hat{\mathcal{R}}(\hat{f})) + \frac{2}{n}\sum_{i=1}^{n} Cov(Y_i, \hat{f}(x_i))$$
$$= E(\hat{\mathcal{R}}(\hat{f})) + \frac{2}{n} trace(S)\sigma_\varepsilon^2.$$

We need estimates for $E(\hat{\mathcal{R}}(\hat{f}))$ and σ_ε^2.

The C_p-statistics

- We estimate the expected training error by the training error we actually observe.
- In order to estimate the noise, one typically takes a "low-bias" model. Practically, one takes a complex model and estimates

$$\hat{\sigma}_{\varepsilon}^2 = \frac{\hat{\mathcal{R}}(\hat{f})}{n - trace(S)}.$$

Resulting estimator:

$$C_p = \hat{\mathcal{R}}(\hat{f}) + \frac{2}{n} trace(s) \hat{\sigma}_{\varepsilon}^2.$$

Comments

- We have only discussed the case where the x_i are fixed, but actually we are interested in the error on *new samples*.
- Nevertheless, the criterion often works, because the *relative performance differences* are predictive for the true performance differences.

Related Methods

- AIC (Akaike Information Criterion)

$$AIC = -2 loglik + \frac{2d}{n}.$$

  More general than the C_p statistic, but identical for the setting discussed.
- BIC (Bayesian Information Criterion)

$$BIC = -2 loglik + (\log n)d.$$

  For our setting:

$$BIC = \frac{n}{\sigma^2}\left(\hat{\mathcal{R}}(\hat{f}) + (\log n)\frac{d}{n}\right).$$

Summary

- Free parameters and other model choices cannot be chosen based on the training error alone.
- Cross-validation as a generic method.
- Optimism-correcting criteria for certain models.
- **Never rely on performance measures on data you have used in the learning process.**