IEOR 165 – Lecture 9
Cross-Validation

1 Cross-Validation

Cross-validation is a data-driven approach that is used to choose tuning parameters for regression. The choice of the λ is an example of a tuning parameter that needs to be chosen in order to use Ridge Regression or Lass Regression. The basic idea of cross-validation is to split the data into two parts. The first part of data is used to compute different estimates (where the difference is due to different tuning parameter values), and the second part of data is used to compute a measure of the quality of the estimate. The tuning parameter that has the best computing measure of quality is selected, and then that particular value for the tuning parameter is used to compute an estimate using all of the data. Cross-validation is closely related to the jackknife and bootstrap methods.

1.1 Leave k-Out Cross-Validation

We can describe this method as an algorithm.

data : (x_i, y_i) for $i = 1, \ldots, n$ (measurements)
input : λ_j for $j = 1, \ldots, z$ (tuning parameters)
input : R (repetition count)
input : k (leave-out size)

output: λ^* (cross-validation selected tuning parameter)

for $j \leftarrow 1$ **to** z **do**
  set $e_j \leftarrow 0$;
end

for $r \leftarrow 1$ **to** R **do**
  set $\mathcal{V}$ to be k randomly picked indices from $\mathcal{I} = \{1, \ldots, n\}$;
  for $j \leftarrow 1$ **to** z **do**
    fit model using λ_j and (x_i, y_i) for $i \in \mathcal{I} \setminus \mathcal{V}$;
    compute cross-validation error $e_j \leftarrow e_j + \sum_{i \in \mathcal{V}} (y_i - \hat{y}_i)^2$;
  end
end
set $\lambda^* \leftarrow \lambda_j$ for $j := \arg\min e_j$;

1.2 k-Fold Cross-Validation

We can describe this method as an algorithm.

data : (x_i, y_i) for $i = 1, \ldots, n$ (measurements)
input : λ_j for $j = 1, \ldots, z$ (tuning parameters)
input : k (block sizes)

output: λ^* (cross-validation selected tuning parameter)

for $j \leftarrow 1$ **to** k **do**
  set $e_j \leftarrow 0$;
end
partition $\mathcal{I} = \{1, \ldots, n\}$ into k randomly chosen subsets $\mathcal{V}_r$ for $r = 1, \ldots, k$;

for $r \leftarrow 1$ **to** k **do**
  for $j \leftarrow 1$ **to** z **do**
    fit model using λ_j and (x_i, y_i) for $i \in \mathcal{I} \setminus \mathcal{V}_r$;
    compute cross-validation error $e_j \leftarrow e_j + \sum_{i \in \mathcal{V}_r} (y_i - \hat{y}_i)^2$;
  end
end
set $\lambda^* = \lambda_j$ for $j := \arg\min e_j$;

1.3 Notes on Cross-Validation

There are a few important points to mention.

1. The first point is that the cross-validation error is an estimate of prediction error, which is defined as
$$\mathbb{E}((\hat{y} - y)^2).$$
One issue with cross-validation error (and this issue is shared by jackknife and bootstrap as well), is that these estimates of prediction error must necessarily be biased lower. The intuition is that we are trying to estimate prediction error using data we have seen, but the true prediction error involves data we have not seen.

2. The second point is related, and it is that the typical use of cross-validation is heuristic in nature. In particular, the consistency of an estimate is affected by the use of cross-validation. There are different cross-validation algorithms, and some algorithms can "destroy" the consistency of an estimator. Because these issues are usually ignored in practice, it is important to remember that cross-validation is usually used in a heuristic manner.

3. The last point is that we can never eliminate the need for tuning parameters. Even though cross-validation allows us to pick a λ^* in a data-driven manner, we have introduced new tuning parameters such as k. The reason that cross-validation is considered to be a data-driven approach to choosing tuning parameters is that estimates are usually less sensitive to the choice of cross-validation tuning parameters, though this is not always true.