STAT 425: Introduction to Nonparametric Statistics **Winter 2018**

Lecture 17: Learning Theory: Model Selection

Instructor: Yen-Chi Chen

17.1 Introduction

Model selection is a central topic in statistics and machine learning. The abstract idea is: given a data and a set of possible models, we want to choose the model that 'best' fits the data. Here the 'best' is generally refer to subject to some measure of fitness (e.g., risk function).

It is a topic that occurs in both parametric and nonparametric methods.

Here are some examples about model selection:

Example 1 (density estimation): Given $X_1, \cdots, X_n$ from an unknown distribution P. Now we have three possible models:

$$\begin{aligned}\mathcal{M}_1 &: \text{ P is a normal distribution} \\ \mathcal{M}_2 &: \text{ P is a gamma distribution} \\ \mathcal{M}_3 &: \text{ P is a Cauchy distribution.}\end{aligned}$$

The question is: which model that best fits the data?

Example 2 (regression): In a regression problem with 2 covariates X_1 and X_2. Let the regression function

$$m(x) = \mathbb{E}(Y|X = x).$$

Assume that we consider a linear model (linear regression). Depending on if we allow each covariate to contribute to the regression function, there are 4 possible models:

$$\begin{aligned}\mathcal{M}_1 &: m(x) = \beta_0 \\ \mathcal{M}_2 &: m(x) = \beta_0 + \beta_1 x_1 \\ \mathcal{M}_3 &: m(x) = \beta_0 + \beta_2 x_2 \\ \mathcal{M}_4 &: m(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2.\end{aligned}$$

The model selection is to choose the right model among these 4 candidates.

Example 3 (classification): Assume in a binary classification problem, we observe IID

$$(X_1, Y_1), \cdots, (X_n, Y_n).$$

We apply a decision tree model and let $\mathcal{M}_k$ denotes the collection of decision trees with k leaves. Now we have a sequence of competing models:

$$\mathcal{M}_1, \mathcal{M}_2, \cdots.$$

The question is to ask: which model will be the best one in predicting future class label?

17.2 Cross-Validation

The cross-validation (CV) is a method for selecting the model that has good predictive power over several possible candidates. Essentially, it is a method for estimating the risk of a model so it can be applied to not only the model selection but also the selection of a tuning parameter.

The cross-validation can be viewed as an extension of the data-splitting technique we have discussed in the previous lecture. For simplicity, we consider the example of decision tree and our data is

$$(X_1, Y_1), \cdots, (X_n, Y_n).$$

The goal is to choose the best model (number of leaves) for predicting future labels. Thus, the problem is like Example 3 where the models are

$$\mathcal{M}_1, \mathcal{M}_2, \cdots$$

and each $\mathcal{M}_k$ is a decision tree with k leaves. The measure the prediction accuracy, we use the $0-1$ loss and the risk function becomes a good measure of success. Let

$$R(k) = \min_{c \in \mathcal{M}_k} R(c) = \min_{c \in \mathcal{M}_k} \mathbb{E}(L(c(x), y)).$$

Leave-one-out CV (LOO-CV). The LOO-CV do a data splitting n times and for the j-th time, we use the following training data and validation data:

$$\begin{aligned} \mathcal{D}_{val,j} &= \{(X_j, Y_j)\}, \\ \mathcal{D}_{tr,j} &= \{(X_1, Y_1), \cdots, (X_{j-1}, Y_{j-1}), (X_{j+1}, Y_{j+1}), \cdots, (X_n, Y_n)\}. \end{aligned}$$

Namely, the validation set is just the j-th observation whereas the training set is all but the j-th observation. This is like leaving the j-th observation out of the training set so that is why it is called 'leave-one out'. Let the classifier $\widehat{c}_j$ be the classifier learned from the dataset $\mathcal{D}_{tr,j}$. The estimated risk using the validation set can be simply written as

$$\widehat{R}_j(k) = L(\widehat{c}_j(X_j), Y_j).$$

Because we have $j = 1, \cdots, n$, totally n estimated risk. We will use their average as an estimate of $R(k)$, which can be written as

$$\widehat{R}_{\mathsf{LOO}}(k) = \frac{1}{n} \sum_{j=1}^{n} \widehat{R}_j(k).$$

By applying this method to each $k = 1, \cdots$, we obtain an estimated risk for each model. Then we choose the model $\mathcal{M}_{\widehat{k}}$ by

$$\widehat{k} = \mathsf{argmin}_k \widehat{R}_{\mathsf{LOO}}(k).$$

k-fold CV. Another popular version of CV is the K-fold CV. We randomly split the data into K equal size groups. Each time we leave out one group and use the other K-1 groups to construct our estimator. Then we use the left out group to evaluate the risk. Repeat this procedure many times and take the average as the risk estimator $\widehat{R}(k)$. Figure 17.2 summarizes how to apply a K-fold CV for decision tree.

After computing $\widehat{R}_{\mathsf{K-fold}}(k)$ for each k, we then choose the model $\mathcal{M}_{\widehat{k}}$ such that

$$\widehat{k} = \mathsf{argmin}_k \widehat{R}_{\mathsf{K-fold}}(k).$$

K-fold Cross-Validation for Decision Tree.

1. Randomly split $\mathcal{D} = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ into K groups: $\mathcal{D}_1, \cdots, \mathcal{D}_K$.
2. For ℓ-th group, use $\mathcal{D}_\ell$ as the validation set and treat the others as training set. Namely, construct the estimator $\widehat{c}^{(\ell)}$ using all the data except ℓ-th group.
3. Evaluate the error by
$$\widehat{R}^{(\ell)}(k) = \frac{1}{n_\ell} \sum_{(X_i, Y_i) \in \mathcal{D}_\ell} L(\widehat{c}^{(\ell)}(X_i), Y_i)$$
4. Compute the average error
$$\widehat{R}^*(k) = \frac{1}{K} \sum_{\ell=1}^{K} \widehat{R}^{(\ell)}(k).$$
5. Repeat the above 4 steps N times, leading to N average errors
$$\widehat{R}^{*(1)}(k), \cdots, \widehat{R}^{*(N)}(k).$$
6. Estimate $R(k)$ via
$$\widehat{R}_{\mathsf{K-fold}}(k) = \frac{1}{N} \sum_{m=1}^{N} \widehat{R}^{*(m)}(k).$$

17.3 Penalty

In addition to the cross-validation, another comment approach of model selection is to use the penalty function we are familiar with. The idea is: for each model $\mathcal{M}_k$, we assign a penalty value $\mathcal{P}(k)$ to it such that a complex model has a higher penalty value whereas a simple model has a lower penalty value. And the goal is to choose the model $\mathcal{M}_{\widehat{k}}$ such that

$$\widehat{k} = \mathsf{argmin}_k \widehat{R}_n(k) + \mathcal{P}(k),$$

where $\widehat{R}_n(k) = \mathsf{argmin}_{c \in \mathcal{M}_k} \frac{1}{n} \sum_{i=1}^{n} L(c(X_i), Y_i)$.

When the model is related to the number of variables (such as the problem in example 2), the penalty function is often chosen proportional to the number of variable being used. For instance, when $\mathcal{M}_k$ denotes the models using k variables, the famous AIC criterion[1] uses $\mathcal{P}(k) = \frac{2k}{n}$ as the penalty and the BIC criterion[2] uses $\mathcal{P}(k) = \frac{k \log n}{n}$.

[1] https://en.wikipedia.org/wiki/Akaike_information_criterion
[2] https://en.wikipedia.org/wiki/Bayesian_information_criterion