MODEL SELECTION

Tuning Parameters

Many machine learning algorithms have parameters that are not determined by the algorithm itself. These parameters influence algorithm performance, and therefore it is important to set them correctly. This problem is called parameter tuning or model selection. A model in this context is a representation of a classifier, regression function, etc.

Examples

- Regularization parameters in regularized logistic regression (λ), ridge regression (λ), and the support vector machine ($C = \frac{1}{\lambda}$).
- Kernel parameters, e.g., Gaussian kernel bandwidth (σ), polynomial kernel degree (p)
- Polynomial degree (p) in polynomial least squares

regression

Parameters usually determine the complexity of a model, and therefore model selection amounts to striking the right balance between overfitting and underfitting.

Estimating Risk

Model selection is an issue in all machine learning. In these notes, we will focus on supervised learning problems in which the performance is measured by a risk.

Let θ denote the parameter(s) to be tuned (θ could be a vector). Let $(x_1, y_1), \ldots, (x_n, y_n)$ be training data, and let $\hat{f}_\theta$ denote the learned model. We would ideally like to solve

$$\min_{\theta} R(\hat{f}_\theta).$$

Of course we don't know R so it must be estimated. If $R(f) = E[L(Y, f(X))]$, a natural estimate is the training error,

$$\hat{R}(\hat{f}) = \frac{1}{n} \sum^{n} L(y_i, \hat{f}(x_i))$$

$$\hat{R}_{TR}(\hat{f}_\theta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}_\theta(x_i)),$$

also known as the <u>apparent error</u> or the <u>resubstitution error</u>. However, if θ determines model complexity, then solving

$$\min_\theta \hat{R}_{TR}(\theta)$$

will select a complex model, leading to overfitting.

Holdout Error

The holdout error estimate is defined as follows. Partition the training data as

$$\underbrace{(x_1,y_1),\ldots,(x_m,y_m)}_{\text{used to fit the model } \hat{f}_\theta},\ \underbrace{(x_{m+1},y_{m+1}),\ldots,(x_n,y_n)}_{\text{used to estimate } R(\hat{f}_\theta)}$$

Assume $\hat{f}_\theta$ is learned using (x_i, y_i), $1 \le i \le m$.

The <u>holdout error estimate</u> is

$$\hat{R}_{HO}(\hat{f}_\theta) = \frac{1}{n-m}\sum^{n} L(y_i, \hat{f}_\theta(x_i))$$

$i=m+1$

Selecting θ by solving

$$\min_{\theta} \hat{R}_{H_0}(\hat{f}_\theta)$$

avoids overfitting. Indeed, suppose

$$(X_1,Y_1), \ldots, (X_n,Y_n) \overset{iid}{\sim} P$$

where P is the joint distribution of (X,Y). I'm using capital letters now to emphasize that these are random variables. Then $\hat{R}_{H_0}(\hat{f}_\theta)$ is an unbiased estimate in the sense that

$$\mathbb{E}_{Z_2|Z_1}\left[\hat{R}_{H_0}(\hat{f}_\theta)\right] = R(\hat{f}_\theta).$$

conditional expectation of
$Z_2 = ((X_{m+1},Y_{m+1}), \ldots, (X_n,Y_n))$ given
$Z_1 = ((X_1,Y_1),\ldots,(X_m,Y_m))$

To see this, observe

$$\mathbb{E}_{Z_2|Z_1}\left[\hat{R}_{H_0}(\hat{f}_\theta)\right] = \frac{1}{n-m}\sum_{i=m+1}^{n} \mathbb{E}_{Z_2|Z_1}\left[L(Y_i,\hat{f}_\theta(X_i))\right]$$

$$= \frac{1}{n-m}\sum_{i=m+1}^{n} \mathbb{E}_{(X_i,Y_i)|Z_1}\left[L(Y_i,\hat{f}_\theta(X_i))\right]$$

$$= \frac{1}{n-m} \sum_{i=m+1}^{n} R(\hat{f}_\theta)$$

since $(X_i, Y_i) \sim P$ and $\hat{f}_\theta$ is fixed given Z_1.

Is the training error also unbiased? What does this even mean? To ask whether

$$\mathbb{E}_Z[\hat{R}_{TR}(\hat{f}_\theta)] = R(\hat{f}_\theta),$$

where

$$Z = ((X_1, Y_1), \ldots, (X_n, Y_n)),$$

doesn't even make sense because the LHS is deterministic and the RHS is random. Instead, we could ask whether

$$\mathbb{E}_Z[\hat{R}_{TR}(\hat{f}_\theta)] = \mathbb{E}_Z[R(\hat{f}_\theta)].$$

So is this true? No, because

$$\mathbb{E}_Z[\hat{R}_{TR}(\hat{f}_\theta)] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_Z[L(Y_i, \hat{f}_\theta(X_i))]$$

and

$$\mathbb{E}_Z[L(Y_i, \hat{f}_\theta(X_i))] \neq \mathbb{E} \quad [L(Y', \hat{f}(X'))]$$

$$\mathbb{E}_Z\left[L(y_i, \hat{f}_\theta(x_i))\right] \neq \mathbb{E}_{Z,(x',y')}\left[L(y', \hat{f}_\theta(x'))\right]$$

$$= \mathbb{E}_Z\left[R(\hat{f}_\theta)\right]$$

where (x', y') is an independent draw from $\mathbb{P}$.

Cross Validation

If data are scarce, the holdout error has the disadvantage that it limits the training examples that are used for learning the model. Cross-validation attempts to improve the sample size used for model fitting while still estimating the error accurately.

Let K be an integer, $1 \le K \le n$. Let $I_1, \dots, I_K$ be a partition of $\{1, 2, \dots, n\}$ such that $|I_j| \approx \frac{n}{K}$ $\forall j$.

Example: $n = 10$, $K = 3$

$I_1 = \{2, 6, 7\}$

$I_2 = \{1, 3, 5, 10\}$

$I_3 = \{4, 8, 9\}$

Define

$$\hat{f}_\theta^{(j)} := \text{model based on } \{(x_i, y_i)\}_{i \notin I_j}$$

and

$$\hat{R}^{(j)}(\hat{f}_\theta^{(j)}) = \frac{1}{|I_j|} \sum_{i \in I_j} L(y_i, \hat{f}_\theta^{(j)}(x_i))$$

Each $\hat{R}^{(j)}$ is like a different holdout estimate.

The K-fold cross-validation error estimate is

$$\hat{R}_{cv}(\hat{f}_\theta) = \frac{1}{K} \sum_{j=1}^{K} \hat{R}^{(j)}(\hat{f}_\theta^{(j)}).$$

Remarks

1. Common choices of K are 5, 10, and n. When $K=n$ it's called leave-one-out cross-validation (LOOCV).

2. To reduce the variance of the estimate, and if you have the time/resources, it is good to compute several CV estimates based on different random partitions and average them.

3. In classification, the sets I_k should be chosen so that the proportions of different classes in each fold are the same as in the full sample.

The Bootstrap

Let $B \geq 1$ be an integer. For $b = 1, \ldots, B$, let I_b be a subset of $\{1, 2, \ldots, n\}$ of size n obtained by sampling with replacement.

Example $n = 6$

$$I_1 = \{3, 4, 5, 4, 1, 2\}$$

$$I_2 = \{1, 2, 6, 6, 2, 5\}$$

Define

$$\hat{f}_\theta^{(b)} = \text{model based on } \{(x_i, y_i)\}_{i \in I_b}$$

and

$$\hat{R}^{(b)}(\hat{f}_\theta^{(b)}) = \frac{1}{n - |I_b|} \sum_{i \notin I_b} L(y_i, \hat{f}_\theta^{(b)}(x_i)).$$

The bootstrap error estimate is

$$\hat{R}_{BS}(\hat{f}_\theta) = \frac{1}{B} \sum^{B} \hat{R}^{(b)}(\hat{f}_\theta^{(b)})$$

$$\hat{R}_{BS}(\hat{f}_\theta) = \frac{1}{B}\sum_{b=1}^{B} \hat{R}^{(b)}(\hat{f}_\theta^{(b)})$$

Remarks

1. The larger B, the better. $B = 200$ is a recommended minimum. That's a lot of training, so the bootstrap can be computationally demanding.
2. $\hat{R}_{BS}$ tends to be pessimistic, so it is common to combine the bootstrap and training error estimates. A recommendation is

$$0.632\,\hat{R}_{BS}(\hat{f}_\theta) + 0.368\,\hat{R}_{TR}(\hat{f}_\theta),$$

called the "0.632 bootstrap."

3. The balanced bootstrap chooses $I_1, \dots, I_B$ such that each $i \in \{1, 2, \dots, n\}$ occurs exactly n times.
4. Reference: Efron and Tibshirani, An Introduction to the Bootstrap.

$\boxed{\text{Final Fitting}}$

With all of the above approaches, once θ is set,

$\hat{f}_B$ is recomputed using all of the training data.