Model Assessment and Selection

Steve Avsec

Illinois Institute of Technology

February 12, 2024

Overview

A Note About Metrics

Regression:

- $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - \hat{f}(\mathbf{X})\|_2.$
- $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - \hat{f}(\mathbf{X})\|_1.$
- $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - \hat{f}(\mathbf{X})\|_p.$
- $L(\mathbf{X}, \textbf{text}) = \sum_{j=1}^{N} \frac{|y_j - \hat{f}(\mathbf{x})|}{|y_j|}$

A Note About Metrics

Regression:

- $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - \hat{f}(\mathbf{X})\|_2$.
- $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - \hat{f}(\mathbf{X})\|_1$.
- $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - \hat{f}(\mathbf{X})\|_p$.
- $L(\mathbf{X}, \mathbf{text}) = \sum_{j=1}^{N} \frac{|y_j - \hat{f}(\mathbf{x})|}{|y_j|}$

Classification:

- $L(\mathbf{X}, \mathbf{y}) = \left|\chi_{\hat{f}(\mathbf{X})=\mathbf{y}}\right|$
- $L(\mathbf{X}, \mathbf{y}) = -2\log(\hat{p}_{\mathbf{y}}(\mathbf{X}))$

Errors

Data has "true structure"

$$\mathbf{y} = f(\mathbf{X}) + \varepsilon$$

ε_j are iid, $E[\varepsilon_j] = 0$, $\text{var}(\varepsilon_j) = \sigma^2$.

Errors

Data has ”true structure”

$$\mathbf{y} = f(\mathbf{X}) + \varepsilon$$

ε_j are iid, $E[\varepsilon_j] = 0$, $\text{var}(\varepsilon_j) = \sigma^2$.
Let $\hat{f}$ be our trained approximation to f. Given a training data set $\mathcal{T}$, define the out-of-sample test error by:

$$\text{err}_{\mathcal{T}} = E[L(\mathbf{X}, \mathbf{y})|\mathcal{T}]$$

Errors

Data has "true structure"

$$\mathbf{y} = f(\mathbf{X}) + \varepsilon$$

ε_j are iid, $E[\varepsilon_j] = 0$, $\text{var}(\varepsilon_j) = \sigma^2$.
Let $\hat{f}$ be our trained approximation to f. Given a training data set $\mathcal{T}$, define the out-of-sample test error by:

$$\text{err}_{\mathcal{T}} = E[L(\mathbf{X}, \mathbf{y})|\mathcal{T}]$$

Define the expected prediction error by (now taking expectation over possible training sets)

$$\text{err} = E[L(\mathbf{X}, \mathbf{y})] = E[\text{err}_{\mathcal{T}}]$$

Bias-Variance Decomposition

For our trained model $\hat{f}$, define its bias as

$$\text{bias}(\hat{f})(\mathbf{x}) = E[\hat{f}(\mathbf{x}) - f(\mathbf{x})]$$

Bias-Variance Decomposition

For our trained model $\hat{f}$, define its bias as

$$\text{bias}(\hat{f})(\mathbf{x}) = E[\hat{f}(\mathbf{x}) - f(\mathbf{x})]$$

Suppose $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - f(\mathbf{X})\|_2$. Then

$$\begin{aligned}
\text{Err}(\mathbf{x}_0) &= E[(Y - \hat{f}(\mathbf{x}_0))^2] \\
&= \sigma^2 + E[f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0)]^2 + E[(\hat{f}(\mathbf{x}_0) - E[\hat{f}(\mathbf{x}_0)]^2]^2 \\
&= \sigma^2 + \text{Bias}(\hat{f})(\mathbf{x}_0) + \text{Var}(\hat{f})(\mathbf{x}_0)
\end{aligned}$$

Bias-Variance Decomposition

For our trained model $\hat{f}$, define its bias as

$$\text{bias}(\hat{f})(\mathbf{x}) = E[\hat{f}(\mathbf{x}) - f(\mathbf{x})]$$

Suppose $L(\mathbf{X}, \mathbf{y}) = \|\mathbf{y} - f(\mathbf{X})\|_2$. Then

$$\begin{aligned}
\text{Err}(\mathbf{x}_0) &= E[(Y - \hat{f}(\mathbf{x}_0))^2] \\
&= \sigma^2 + E[f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0)]^2 + E[(\hat{f}(\mathbf{x}_0) - E[\hat{f}(\mathbf{x}_0)]^2]^2 \\
&= \sigma^2 + \text{Bias}(\hat{f})(\mathbf{x}_0) + \text{Var}(\hat{f})(\mathbf{x}_0)
\end{aligned}$$

The Bias-Variance Tradeoff: if the total prediction error is fixed, a model with low bias will have high variance.

Optimism

Define the training error by

$$\overline{\text{err}} = \frac{1}{N} L(\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}})$$

where $(\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}})$ denotes the training data set.

Optimism

Define the training error by

$$\overline{\text{err}} = \frac{1}{N} L(\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}})$$

where $(\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}})$ denotes the training data set.
Define the in-sample error by

$$\text{err}_{\text{in}} = \frac{1}{N} E_{Y^0}[L(\mathbf{X}_{\mathcal{T}}, Y^0)]$$

where the input training samples are fixed, but the *outcomes* are resampled.

Optimism

Define the training error by

$$\overline{\text{err}} = \frac{1}{N} L(\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}})$$

where $(\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}})$ denotes the training data set.
Define the in-sample error by

$$\text{err}_{\text{in}} = \frac{1}{N} E_{Y^0}[L(\mathbf{X}_{\mathcal{T}}, Y^0)]$$

where the input training samples are fixed, but the *outcomes* are resampled.
The *optimism* of the training error is defined as the difference of these

$$\text{op} = \text{err}_{\text{in}} - \overline{\text{err}}$$

AIC

For some commonly used loss functions, it generally holds that

$$\omega = E_{\mathcal{T}}[\text{op}] = \frac{2}{N}\text{Cov}\left(\mathbf{y}, \hat{f}(\mathbf{X})\right)$$

AIC

For some commonly used loss functions, it generally holds that

$$\omega = E_{\mathcal{T}}[\text{op}] = \frac{2}{N}\text{Cov}\left(\mathbf{y}, \hat{f}(\mathbf{X})\right)$$

For linear functions, it is not hard to show that

$$\text{Cov}\left(\mathbf{y}, \hat{f}(\mathbf{X})\right) = d\sigma$$

where d is the dimension of the input (e.g. X is N-by-d).

AIC

For some commonly used loss functions, it generally holds that

$$\omega = E_{\mathcal{T}}[\text{op}] = \frac{2}{N}\text{Cov}\left(\mathbf{y}, \hat{f}(\mathbf{X})\right)$$

For linear functions, it is not hard to show that

$$\text{Cov}\left(\mathbf{y}, \hat{f}(\mathbf{X})\right) = d\sigma$$

where d is the dimension of the input (e.g. X is N-by-d). This leads to

$$E_{\mathbf{Y}^0}[\text{err}_{\text{in}}] = E_{\mathcal{T}}[\overline{\text{err}}] + \frac{2d\sigma}{N}$$

This is equivalent in the linear case to the Akaike information criterion

$$\text{AIC} = -\frac{2}{N}\text{loglik} + \frac{2d\sigma}{N}$$

Effective Number of Parameters

Recall that for plain old linear regression, we had

$$\mathbf{c} = V\Sigma^{\dagger}U^{t}\mathbf{y}$$

and thus

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{X}\mathbf{c} \\ &= U\Sigma V^{t}V\Sigma^{\dagger}U^{t}\mathbf{y} \\ &= UPU^{t}\mathbf{y}\end{aligned}$$

where P is a diagonal matrix with d 1s down the diagonal and the rest 0.

Effective Dimension

To recover d, we use the trace:

$$\mathrm{Tr}(UPU^t) = d$$

and in general

$$\mathrm{df}(S) = \mathrm{Tr}(S)$$

where $\hat{\mathbf{y}} = S\mathbf{y}$.

Tying It Up

We have

$$\mathrm{AIC}(\alpha) = -\frac{2}{N}\mathrm{loglik} + \frac{2\mathrm{df}_\alpha \sigma}{N}$$

where α denotes a collection of hyperparameters for your model.

Tying It Up

We have

$$\mathrm{AIC}(\alpha) = -\frac{2}{N}\mathrm{loglik} + \frac{2\mathrm{df}_{\alpha}\sigma}{N}$$

where α denotes a collection of hyperparameters for your model.

Find the set of hyperparameters $\hat{\alpha}$ such that $\mathrm{AIC}(\alpha)$ is *minimized* ensures you balance between best fit (loglik close to 0) and fewest effective parameters (df).

Cross Validation

In general, we are looking to compute the expected prediction error

$$\text{err} = E[L(\mathbf{X}, \mathbf{y})|(T)]$$

Cross Validation

In general, we are looking to compute the expected prediction error

$$\mathrm{err} = E[L(\mathbf{X}, \mathbf{y})|(T)]$$

Steps:

1. Randomly split data into k roughly equal-sized partitions. Let

$$\kappa : [1, \ldots, N] \rightarrow [1, \ldots, K]$$

denote the indexing function (which sample goes to which partition).

K-Fold Cross Validation Steps

2. Donote by

$$\mathrm{CV}(\hat{f}, \alpha) = \frac{1}{N} \sum_{j=1}^{N} L(\hat{f}^{-\kappa(j)}(\mathbf{x}_j), y_j)$$

where $\hat{f}^{-\kappa(j)}$ denote the model trained on the data minus the $\kappa(j)$th partition.

K-Fold Cross Validation Steps

2. Donote by

$$CV(\hat{f}, \alpha) = \frac{1}{N} \sum_{j=1}^{N} L(\hat{f}^{-\kappa(j)}(\mathbf{x}_j), y_j)$$

   where $\hat{f}^{-\kappa(j)}$ denote the model trained on the data minus the $\kappa(j)$th partition.

3. Minimize $CV(\hat{f}, \alpha)$ with respect to α (usually via something like grid search).

Bias And Variance and K

If $K = N$, is bias or variance greater? Why?

Bias And Variance and K

If $K = N$, is bias or variance greater? Why?

If $K = 1$, is bias or variance greater? Why?

Bias And Variance and K

If $K = N$, is bias or variance greater? Why?

If $K = 1$, is bias or variance greater? Why?

Typical values for K: 5 or 10.

What Not To Do

DO NOT:

What Not To Do

DO NOT:

1. Do any kind of variable selection on the whole data set before doing K-fold CV. (See Section 7.10.2 of ESL for a great example.)

What Not To Do

DO NOT:

1. Do any kind of variable selection on the whole data set before doing K-fold CV. (See Section 7.10.2 of ESL for a great example.)

2. Apply CV to high-dimensional problems without being very careful about retraining from scratch on each fold.

What Not To Do

DO NOT:

1. Do any kind of variable selection on the whole data set before doing K-fold CV. (See Section 7.10.2 of ESL for a great example.)
2. Apply CV to high-dimensional problems without being very careful about retraining from scratch on each fold.
3. Apply CV to time-stamped data unless you have a *really* good understanding of distributional shifts in the data first.

What Not To Do

DO NOT:

1. Do any kind of variable selection on the whole data set before doing K-fold CV. (See Section 7.10.2 of ESL for a great example.)

2. Apply CV to high-dimensional problems without being very careful about retraining from scratch on each fold.

3. Apply CV to time-stamped data unless you have a *really* good understanding of distributional shifts in the data first.

Steps

1. Pull B samples (usually a large number of ”small” samples) from the training data (with replacement).

Steps

1. Pull B samples (usually a large number of "small" samples) from the training data (with replacement).

2. Train models for each of the samples (call these $\hat{f}^b$).

Steps

1. Pull B samples (usually a large number of "small" samples) from the training data (with replacement).
2. Train models for each of the samples (call these $\hat{f}^b$).
3. Compute the error

$$\mathrm{Err}_{\mathrm{boot}}(\alpha) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{|C_j|}\sum_{b\in C_j} L(\hat{f}^b_\alpha(\mathbf{X}), \mathbf{y})$$

Some Rules of Thumb

- AIC tends to work better for linear, regularized and models where computing S is not tremendously onerous.

Some Rules of Thumb

- AIC tends to work better for linear, regularized and models where computing S is not tremendously onerous.
- K-fold CV and Bootstrapping tends to work better for tree-based models and other situations where computing S is difficult.

Some Rules of Thumb

- AIC tends to work better for linear, regularized and models where computing S is not tremendously onerous.
- K-fold CV and Bootstrapping tends to work better for tree-based models and other situations where computing S is difficult.
- Both K-fold CV and Bootstrapping can lead to underestimating true errors (particularly in tree-based models) due to tuning parameters entirely in one sample.

Some Rules of Thumb

- AIC tends to work better for linear, regularized and models where computing S is not tremendously onerous.
- K-fold CV and Bootstrapping tends to work better for tree-based models and other situations where computing S is difficult.
- Both K-fold CV and Bootstrapping can lead to underestimating true errors (particularly in tree-based models) due to tuning parameters entirely in one sample.
- Trends and drift in data over time can take a lot of this tuning sideways. Think carefully about what might be driving shifts in your data and act appropriately.