Assessing Performance

CSE 446: Machine Learning
Emily Fox
University of Washington
January 11, 2017

Make predictions, get $, right??

Model + algorithm → fitted function

Predictions → decisions → outcome
Or, how much am I losing?

Example: Lost $ due to inaccurate listing price
- Too low → low offers
- Too high → few lookers + no/low offers

How much am I losing compared to perfection?

Perfect predictions: Loss = 0
My predictions: Loss = ???

Measuring loss

Loss function:
\[ L(y, f_\hat{w}(x)) \]

Cost of using \( \hat{w} \) at \( x \) when \( y \) is true

actual value

\( \hat{f}(x) = \text{predicted value } \hat{y} \)

Examples: (assuming loss for underpredicting = overpredicting)

Absolute error: \( L(y, f_\hat{w}(x)) = |y - f_\hat{w}(x)| \)
Squared error: \( L(y, f_\hat{w}(x)) = (y - f_\hat{w}(x))^2 \)
“Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.” George Box, 1987.

Assessing the loss
Assessing the loss

**Part 1: Training error**

Define training data

![Graph](image)

- **y** (price ($) vs. square feet (sq.ft.))
- **x** (square feet (sq.ft.))
Define training data

Example:
Fit quadratic to minimize RSS

\[ \hat{w} \text{ minimizes RSS of training data} \]
Compute training error

1. Define a loss function \( L(y, f_{\mathbf{w}}(x)) \)
   - E.g., squared error, absolute error,…

2. Training error
   \[ \text{Training error} = \text{avg. loss on houses in training set} \]
   \[ = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_{\mathbf{w}}(x_i)) \]

Example:
Use squared error loss \((y - f_{\mathbf{w}}(x))^2\)

Training error \((\mathbf{w}) = \frac{1}{N} \times \left( \frac{(\text{price}_{\text{train} 1} - f_{\mathbf{w}}(\text{sq.ft.}_{\text{train} 1}))^2}{\text{price}_{\text{train} 2} - f_{\mathbf{w}}(\text{sq.ft.}_{\text{train} 2}))^2} + \frac{(\text{price}_{\text{train} 3} - f_{\mathbf{w}}(\text{sq.ft.}_{\text{train} 3}))^2}{\ldots \text{include all training houses}} \right) \]
Training error vs. model complexity

Error vs. Model complexity

- Error on the Y-axis
- Model complexity on the X-axis

The graph shows the relationship between model complexity and training error for different price points and square footage (sq.ft.) values.
Training error vs. model complexity

Error vs. Model complexity

Training error vs. model complexity

Error vs. Model complexity
Is training error a good measure of predictive performance?

How do we expect to perform on a new house?
Is training error a good measure of predictive performance?

Is there something particularly bad about having $x_t$ sq.ft.?

**Issue:**
Training error is overly optimistic... $\hat{w}$ was fit to training data

Small training error $\Rightarrow$ good predictions unless training data includes everything you might ever see
Assessing the loss

Part 2: Generalization (true) error

Generalization error

Really want estimate of loss over all possible (🏠, $) pairs

Lots of houses in neighborhood, but not in dataset
Distribution over houses

In our neighborhood, houses of what \# sq.ft. (\above) are we likely to see?

Distribution over sales prices

For houses with a given \# sq.ft. (\above), what house prices $ are we likely to see?

For fixed \# sq.ft.
Generalization error definition

Really want estimate of loss over all possible \((\text{house}, \$)\) pairs

Formally:

\[
\text{generalization error} = \mathbb{E}_{x,y}[L(y, f_\mathbf{w}(x))]
\]

average over all possible \((x,y)\) pairs weighted by how likely each is

fit using training data

Generalization error vs. model complexity

Error

Model complexity

price ($)

square feet (sq.ft.)

\(y\)

\(x\)
Generalization error vs. model complexity

Error

Model complexity

y

price ($) 

x

square feet (sq.ft.)

Generalization error vs. model complexity

Error

Model complexity

y

price ($) 

x

square feet (sq.ft.)
Generalization error vs. model complexity

Error vs. Model complexity

Error vs. Model complexity
Generalization error vs. model complexity

Assessing the loss
Part 3: Test error
Approximating generalization error

Wanted estimate of loss over all possible \((\text{home}, \$)\) pairs

Approximate by looking at houses not in training set

Forming a test set

Hold out some \((\text{home}, \$)\) that are *not* used for fitting the model
Forming a test set

Hold out some \( (x, y) \) that are *not* used for fitting the model

Proxy for “everything you might see”

Test set

Compute test error

Test error

\[
\text{Test error} = \text{avg. loss on houses in test set} = \frac{1}{N_{\text{test}}} \sum_{i \in \text{test set}} L(y_i, f_\theta(x_i))
\]

# test points

fit using training data

has never seen test data!
Example:
As before, fit quadratic to training data

\[
\hat{w} \text{ minimizes RSS of training data}
\]

Example:
As before, use squared error loss \((y - f_{\hat{w}}(x))^2\)

\[
\text{Test error } (\hat{w}) = \frac{1}{N_{\text{test}}} * \\
\left[ (\text{\$test}_1 - f_{\hat{w}}(\text{sq.ft. test}_1))^2 \right. \\
\left. + (\text{\$test}_2 - f_{\hat{w}}(\text{sq.ft. test}_2))^2 \right. \\
\left. + (\text{\$test}_3 - f_{\hat{w}}(\text{sq.ft. test}_3))^2 \right. \\
\ldots \text{ include all test houses}\]
Training, true, & test error vs. model complexity

Overfitting if:

Training/test split
Training/test splits

Training set | Test set

how many? vs. how many?

Too few $\rightarrow$ $\hat{w}$ poorly estimated
Too few $\rightarrow$ test error bad approximation of generalization error

Typically, just enough test points to form a reasonable estimate of generalization error

If this leaves too few for training, other methods like cross validation (will see later...)
3 sources of error +
the bias-variance tradeoff

3 sources of error

In forming predictions, there are 3 sources of error:

1. Noise
2. Bias
3. Variance
Data inherently noisy

\[ y_i = f_{\text{true}}(x_i) + \epsilon_i \]

Irreducible error

Bias contribution

Assume we fit a constant function
**Bias contribution**

Over all possible size N training sets, what do I expect my fit to be?

$$f_{\hat{w}}(\text{train1})$$

$$f_{\hat{w}}(\text{train2})$$

$$f_{\hat{w}}(\text{train3})$$

$$f_w(\text{true})$$

**Bias contribution**

$$\text{Bias}(x) = f_w(\text{true})(x) - f_w(x)$$

Is our approach flexible enough to capture $$f_w(\text{true})$$? If not, error in predictions.

**Is our approach flexible enough to capture**

$$f_w(\text{true})$$?

**If not, error in predictions.**

- **low complexity**
- **high bias**
Variance contribution

How much do specific fits vary from the expected fit?

\[ f_{\hat{w}(\text{train}1)}, f_{\hat{w}(\text{train}2)}, f_{\hat{w}(\text{train}3)} \]
Variance contribution

How much do specific fits vary from the expected fit?

Can specific fits vary widely? If so, erratic predictions

Variance of high-complexity models

Assume we fit a high-order polynomial
Variance of high-complexity models

Assume we fit a high-order polynomial

\[ f_{\hat{w}}(\text{train1}) \]
\[ f_{\hat{w}}(\text{train2}) \]
\[ f_{\hat{w}}(\text{train3}) \]

high complexity \rightarrow high variance

CSE 446: Machine Learning
Bias of high-complexity models

Bias-variance tradeoff
Error vs. amount of data

Summary of assessing performance
What you can do now...

- Describe what a loss function is and give examples
- Contrast training, generalization, and test error
- Compute training and test error given a loss function
- Discuss issue of assessing performance on training set
- Describe tradeoffs in forming training/test splits
- List and interpret the 3 sources of avg. prediction error
  - Irreducible error, bias, and variance

More in depth on the 3 sources of errors...

**OPTIONAL**
Accounting for training set randomness

Training set was just a random sample of N houses sold

What if N other houses had been sold and recorded?

generalization error of $\hat{w}(1)$  
generalization error of $\hat{w}(2)$
Accounting for training set randomness

Ideally, want performance averaged over all possible training sets of size $N$

Expected prediction error

$$E_{\text{training set}}[\text{generalization error of } \hat{w}(\text{training set})]$$

- Averaging over all training sets (weighted by how likely each is)
- Parameters fit on a specific training set
Prediction error at target input

Start by considering:
1. Loss at target $x_t$ (e.g. 2640 sq.ft.)
2. Squared error loss $L(y, f_{\hat{w}}(x)) = (y - f_{\hat{w}}(x))^2$

Sum of 3 sources of error

Average prediction error at $x_t$
$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$
Error variance of the model

Average prediction error at $x_t$

$$\sigma^2 = \text{variance}$$

$$y = f_{w(\text{true})}(x) + \epsilon$$

Irreducible error

Bias of function estimator

Average prediction error at $x_t$

$$\sigma^2 = \text{variance}$$
Bias of function estimator

Average estimated function = \( f_{\mathbf{w}}(\mathbf{x}) \)
True function = \( f_{\mathbf{w}}(\mathbf{x}) \)

\[
E_{\text{train}}[f_{\mathbf{w}}(\text{train})](\mathbf{x})
\]

over all training sets of size N

\[
\text{bias}(f_{\mathbf{w}}(\mathbf{x}_t)) = f_{\mathbf{w}}(\mathbf{x}_t) - f_{\bar{\mathbf{w}}}(\mathbf{x}_t)
\]
Bias of function estimator

Average prediction error at $x_t$

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$

Variance of function estimator

Average prediction error at $x_t$

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$
Variance of function estimator

Average prediction error at $x_t$

$$\text{Average prediction error at } x_t = \sigma^2 + [\text{bias}(f_{\bar{w}}(x_t))]^2 + \text{var}(f_{\bar{w}}(x_t))$$

fit on a specific training dataset

$$\text{var}(f_{\bar{w}}(x_t)) = E_{\text{train}}[(f_{\bar{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t))^2]$$

over all training sets of size $N$

deviation of specific fit from expected fit at $x_t$
Why 3 sources of error?
A formal derivation

Deriving expected prediction error

Expected prediction error
  = \( E_{\text{train}} \) [generalization error of \( \hat{w}(\text{train}) \)]
  = \( E_{\text{train}} \) \( E_{x,y} \left[ L(y, f(\hat{w}(\text{train})(x))) \right] \)

1. Look at specific \( x_t \)
2. Consider \( L(y, f(\hat{w}(x))) = (y - f(\hat{w}(x)))^2 \)

Expected prediction error at \( x_t \)
  = \( E_{\text{train},y_t} \left[ (y_t - f(\hat{w}(\text{train})(x_t)))^2 \right] \)
Deriving expected prediction error

Expected prediction error at $x_t$

$$E_{\text{train},y_t}[(y_t - f_{\hat{w}(\text{train})}(x_t))^2]$$

$$= E_{\text{train},y_t}[(y_t - f_{w(\text{true})}(x_t)) + (f_{w(\text{true})}(x_t) - f_{\hat{w}(\text{train})}(x_t))^2]$$

$$= E_{\text{train},y_t}[(y_t - f_{\hat{w}})^2] + 2 E_{\text{train}}[(y_t - f_{\hat{w}})(f_{\hat{w}} - f_{\hat{w}(\text{true})})] + E_{\text{train}}[(f_{\hat{w}} - f_{\hat{w}(\text{true})})^2]$$

$$= \sigma^2 + \text{MSE}(\hat{f})$$

Equating MSE with bias and variance

$$\text{MSE}[f_{\hat{w}(\text{train})}(x_t)]$$

$$= E_{\text{train}}[(f_{w(\text{true})}(x_t) - f_{\hat{w}(\text{train})}(x_t))^2]$$

$$= E_{\text{train}}[(f_{w(\text{true})}(x_t) - f_{\hat{w}}(x_t)) + (f_{\hat{w}}(x_t) - f_{\hat{w}(\text{train})}(x_t))^2]$$

$$= E_{\text{train}}[(f_{\hat{w}} - \hat{f})^2] + 2 E_{\text{train}}[(f_{\hat{w}} - \hat{f})(\hat{f} - f_{\hat{w}(\text{true})})] + E_{\text{train}}[(\hat{f} - f_{\hat{w}(\text{true})})^2]$$

$$= \text{bias}^2(\hat{f}) + \text{var}(\hat{f})$$
Putting it all together

Expected prediction error at $x_t$

\[
\begin{align*}
\text{Expected prediction error} &= \sigma^2 + \text{MSE}[f_w(x_t)] \\
&= \sigma^2 + \left[\text{bias}(f_w(x_t))\right]^2 + \text{var}(f_w(x_t))
\end{align*}
\]

3 sources of error