Data 401
Comparing Models

Dennis Sun

October 5, 2016

1. Review of Last Class
2. How Not to Compare Models: Training Error
3. AIC and BIC
4. Prediction Error and Cross Validation

1. Review of Last Class
2. How Not to Compare Models: Training Error
3. AIC and BIC
4. Prediction Error and Cross Validation

What We Learned

What We Learned

- The possible features we have to consider are not just the features in our data, but also basis expansions and interactions of those features.

What We Learned

- The possible features we have to consider are not just the features in our data, but also basis expansions and interactions of those features.
- The more features we include, the lower the bias of our model, but the higher the variance.

What We Learned

- The possible features we have to consider are not just the features in our data, but also basis expansions and interactions of those features.
- The more features we include, the lower the bias of our model, but the higher the variance.
- We want to include just enough variables to have low bias, but not so many that variance is too high.

Today's Class

Today's Class

- We've evaluated how good a model is by simply making a scatterplot and looking at the fitted model.

Today's Class

- We've evaluated how good a model is by simply making a scatterplot and looking at the fitted model.
- This is only possible because we are effectively only looking at one feature.

Today's Class

- We've evaluated how good a model is by simply making a scatterplot and looking at the fitted model.
- This is only possible because we are effectively only looking at one feature.
- Today, we'll learn strategies to compare models and select one with a good bias-variance tradeoff automatically.

Today's Class

- We've evaluated how good a model is by simply making a scatterplot and looking at the fitted model.
- This is only possible because we are effectively only looking at one feature.
- Today, we'll learn strategies to compare models and select one with a good bias-variance tradeoff automatically.
- Note that we still have to *try* all the different models, which may not be feasible. We'll learn how to efficiently explore the space of possible models next time.

1 Review of Last Class

2 How Not to Compare Models: Training Error

3 AIC and BIC

4 Prediction Error and Cross Validation

Training Error

- How do we determine what is a good model?

Training Error

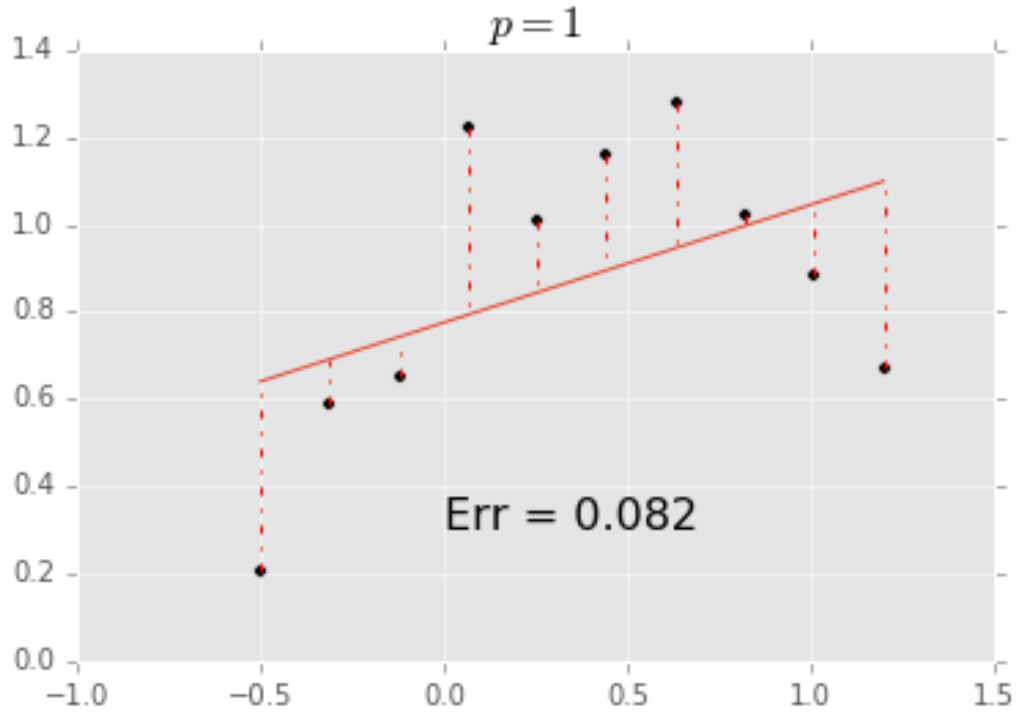
- How do we determine what is a good model?
- One possibility is the training error:

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi})^2.$$

Training Error

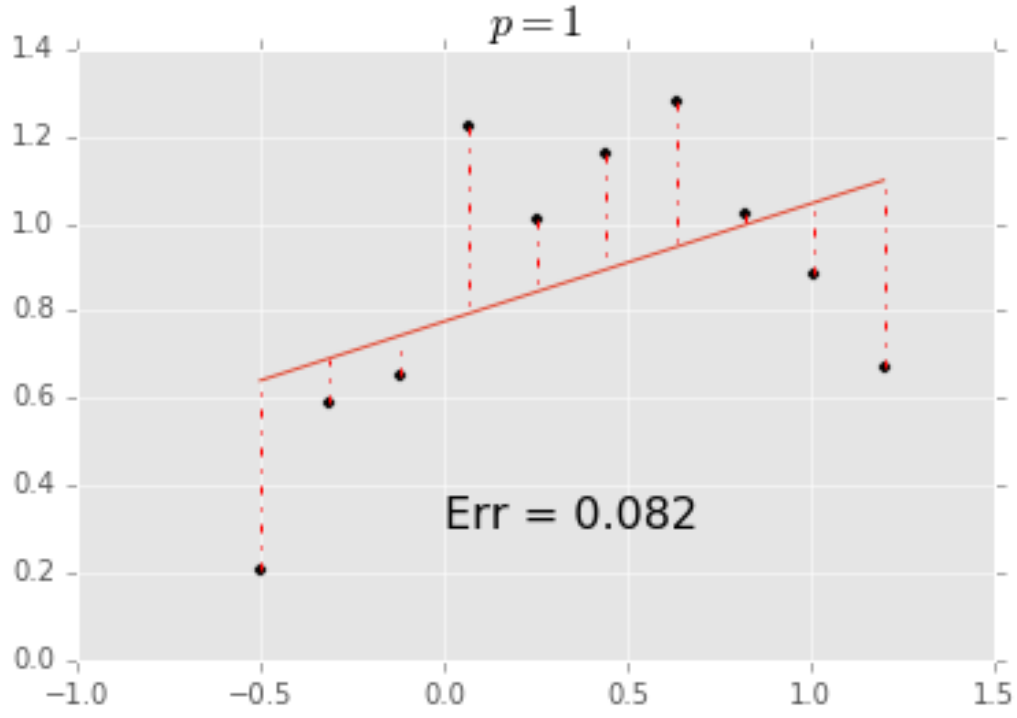
- How do we determine what is a good model?
- One possibility is the training error:

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi})^2.$$



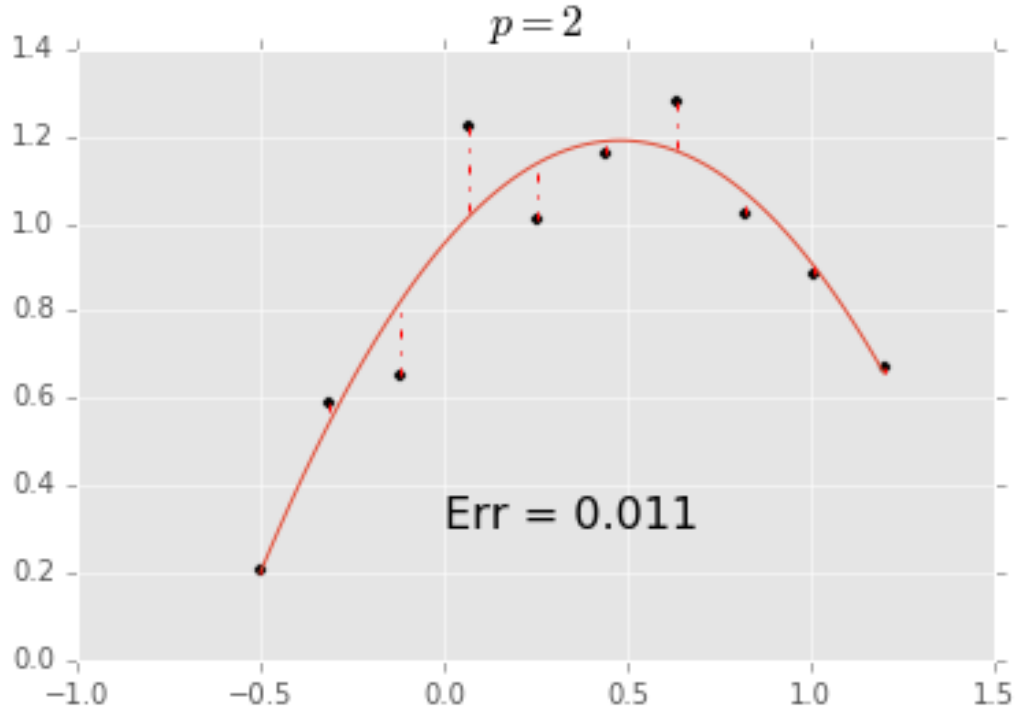
Training Error

Let's calculate the training error from fitting various polynomial models to this data.



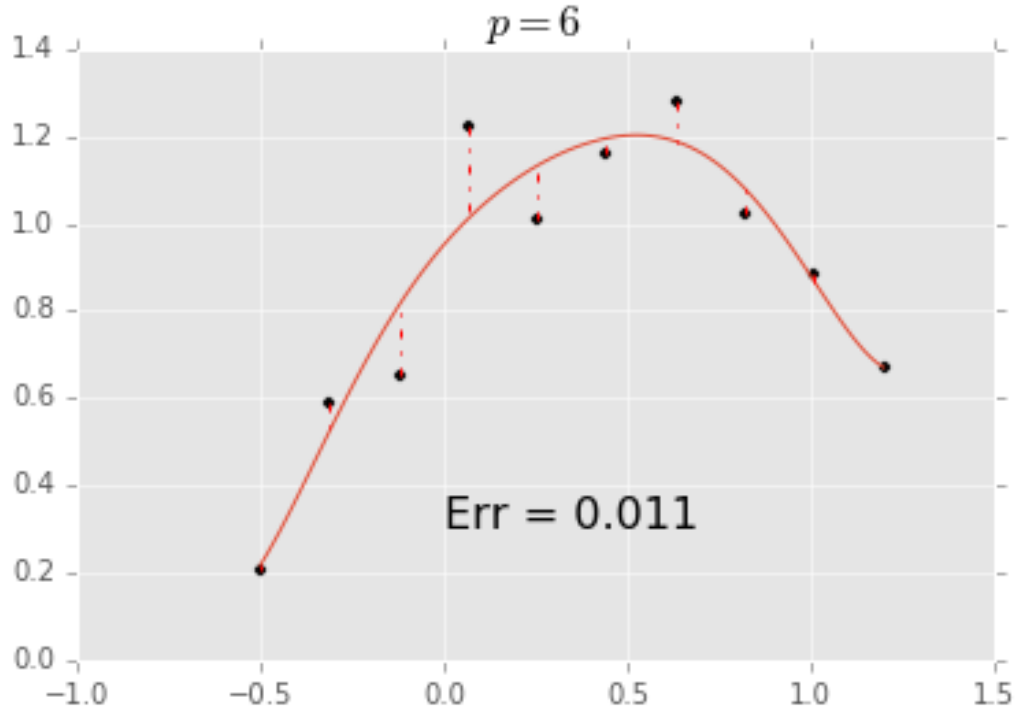
Training Error

Let's calculate the training error from fitting various polynomial models to this data.



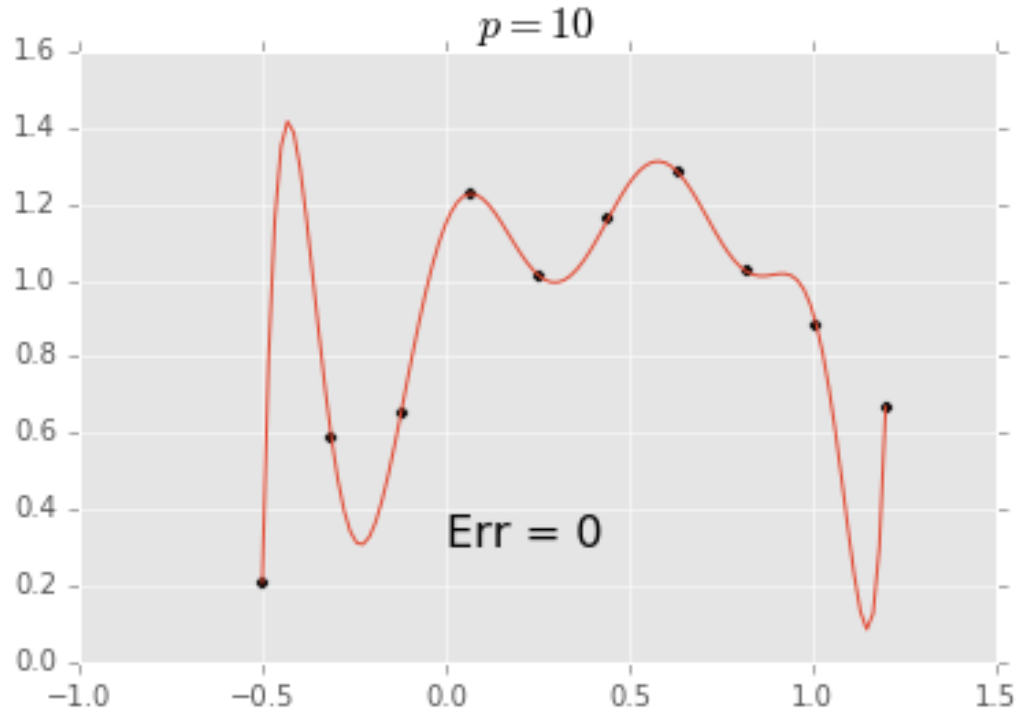
Training Error

Let's calculate the training error from fitting various polynomial models to this data.



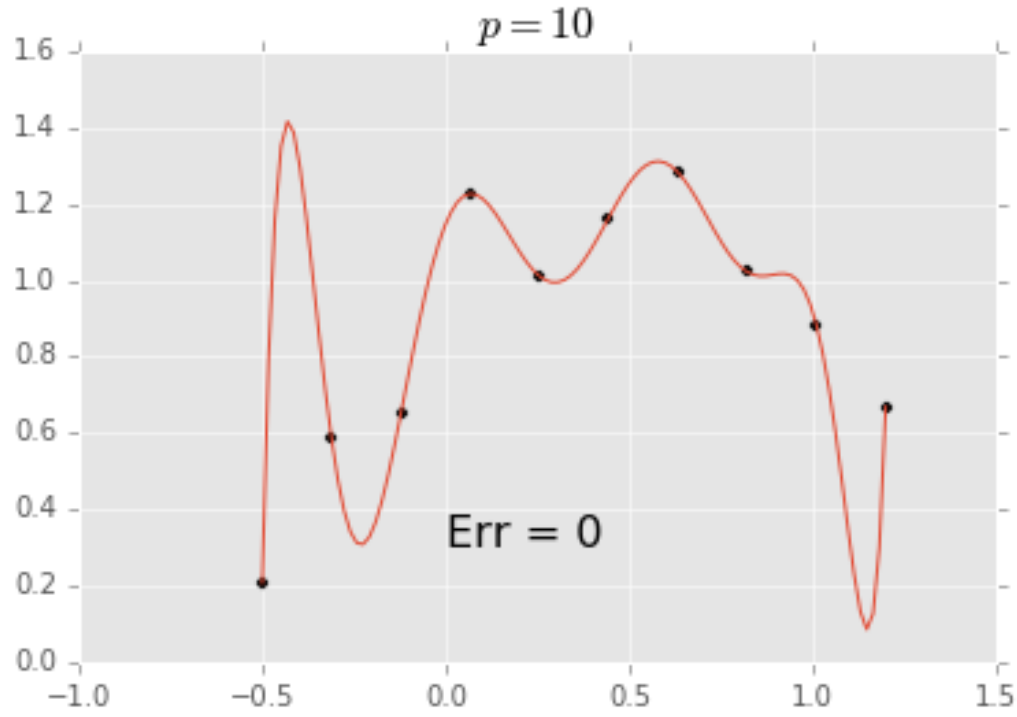
Training Error

Let's calculate the training error from fitting various polynomial models to this data.



Training Error

Let's calculate the training error from fitting various polynomial models to this data.



The training error only goes down as the number of parameters increases. It does not capture the bias-variance tradeoff.

1 Review of Last Class

2 How Not to Compare Models: Training Error

3 AIC and BIC

4 Prediction Error and Cross Validation

AIC

The **Akaike Information Criterion** (AIC) chooses the model with the lowest value of

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi}))^2 + 2p.$$

AIC

The **Akaike Information Criterion** (AIC) chooses the model with the lowest value of

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi}))^2 + 2p.$$

The first term is just the training error, which can only go down as we increase the number of parameters.

AIC

The **Akaike Information Criterion** (AIC) chooses the model with the lowest value of

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi}))^2 + 2p.$$

The first term is just the training error, which can only go down as we increase the number of parameters.

So AIC adds a penalty for the number of parameters p. So if two models have similar training errors, AIC will prefer the one with fewer parameters.

BIC

The **Bayesian Information Criterion** (BIC) chooses the model with the lowest value of

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi}))^2 + p\log(n).$$

BIC

The **Bayesian Information Criterion** (BIC) chooses the model with the lowest value of

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi}))^2 + p\log(n).$$

The only difference between AIC and BIC is that 2 is replaced by $\log(n)$. Will AIC or BIC favor a smaller model more strongly?

BIC

The **Bayesian Information Criterion** (BIC) chooses the model with the lowest value of

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + ... + \hat{\beta}_p x_{pi}))^2 + p\log(n).$$

The only difference between AIC and BIC is that 2 is replaced by $\log(n)$. Will AIC or BIC favor a smaller model more strongly?

BIC is consistent. That is, as $n \to \infty$, it will choose the correct model.

1. Review of Last Class
2. How Not to Compare Models: Training Error
3. AIC and BIC
4. Prediction Error and Cross Validation

Prediction Error

Prediction Error

- Instead of penalizing the number of parameters, we can also optimize for **prediction error**, the error if we were to apply this model to new data.

Prediction Error

- Instead of penalizing the number of parameters, we can also optimize for **prediction error**, the error if we were to apply this model to new data.
- A model with a bad bias-variance tradeoff will have bad prediction error.

Prediction Error

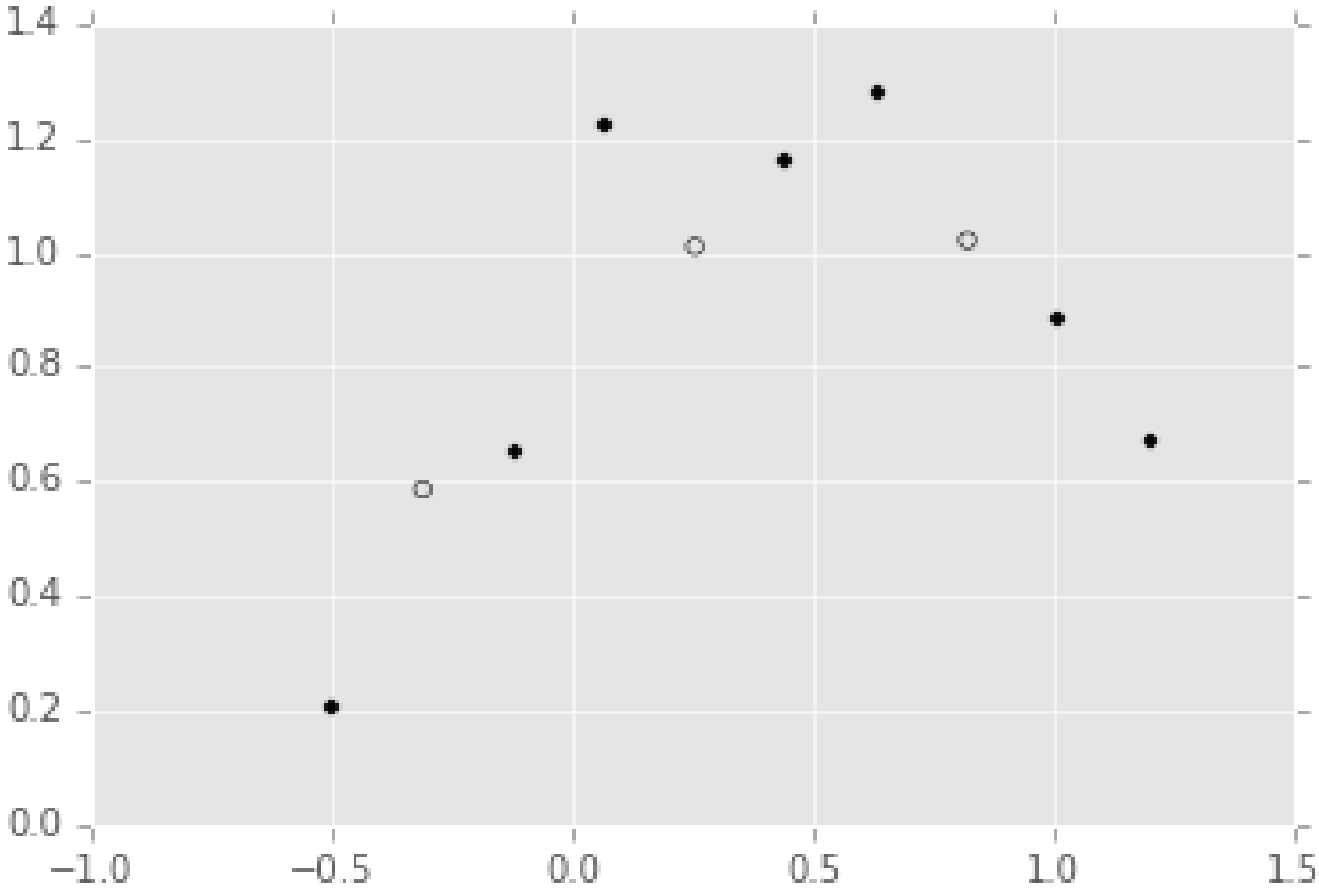
- Instead of penalizing the number of parameters, we can also optimize for **prediction error**, the error if we were to apply this model to new data.
- A model with a bad bias-variance tradeoff will have bad prediction error.
- How do we estimate prediction error?

Prediction Error

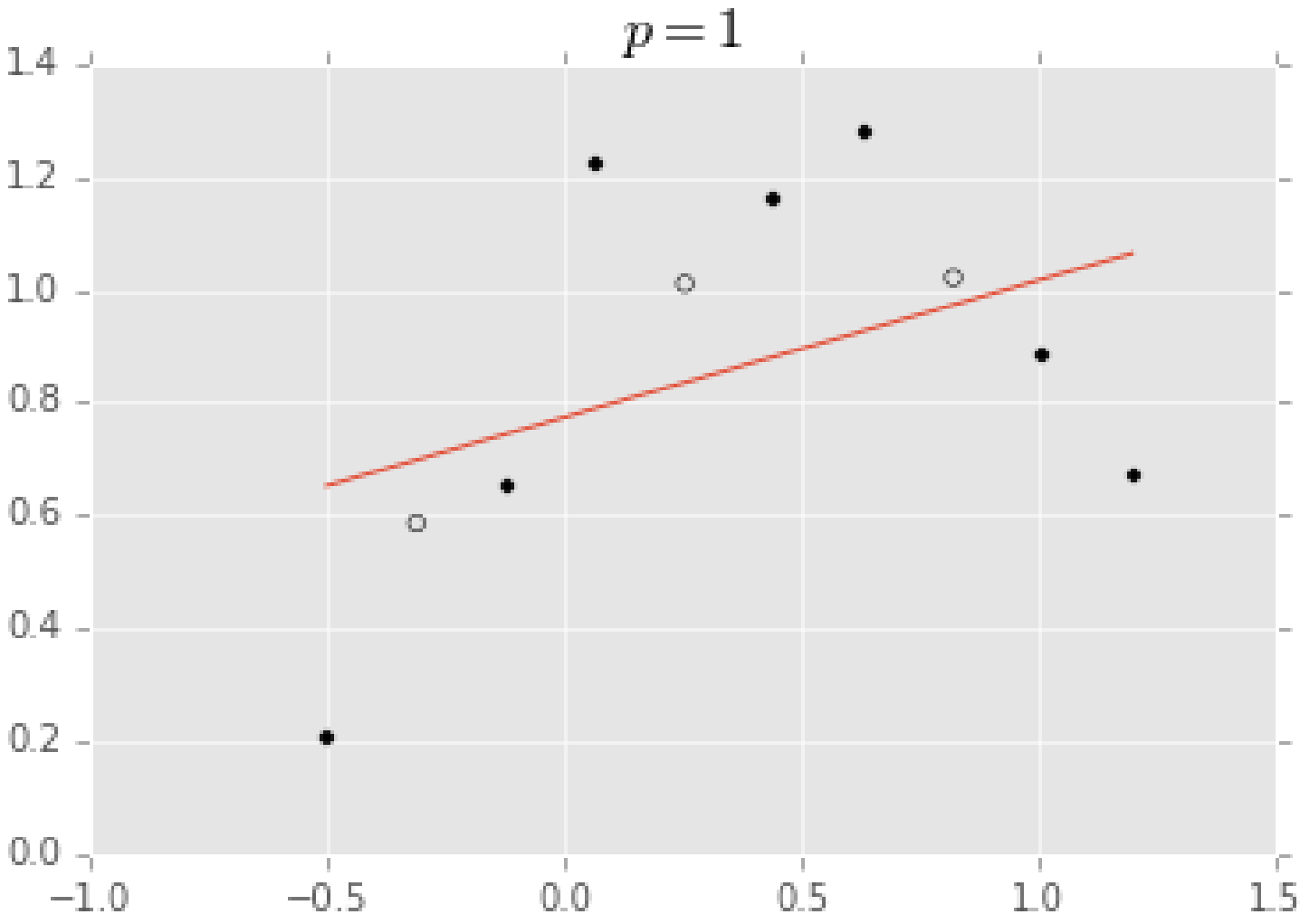
- Instead of penalizing the number of parameters, we can also optimize for **prediction error**, the error if we were to apply this model to new data.
- A model with a bad bias-variance tradeoff will have bad prediction error.
- How do we estimate prediction error?

  Answer: Split the data into training and test sets. Use only the training data to fit the model. Then, we can evaluate the error of our predictions on the test set. This is called the **test error**.

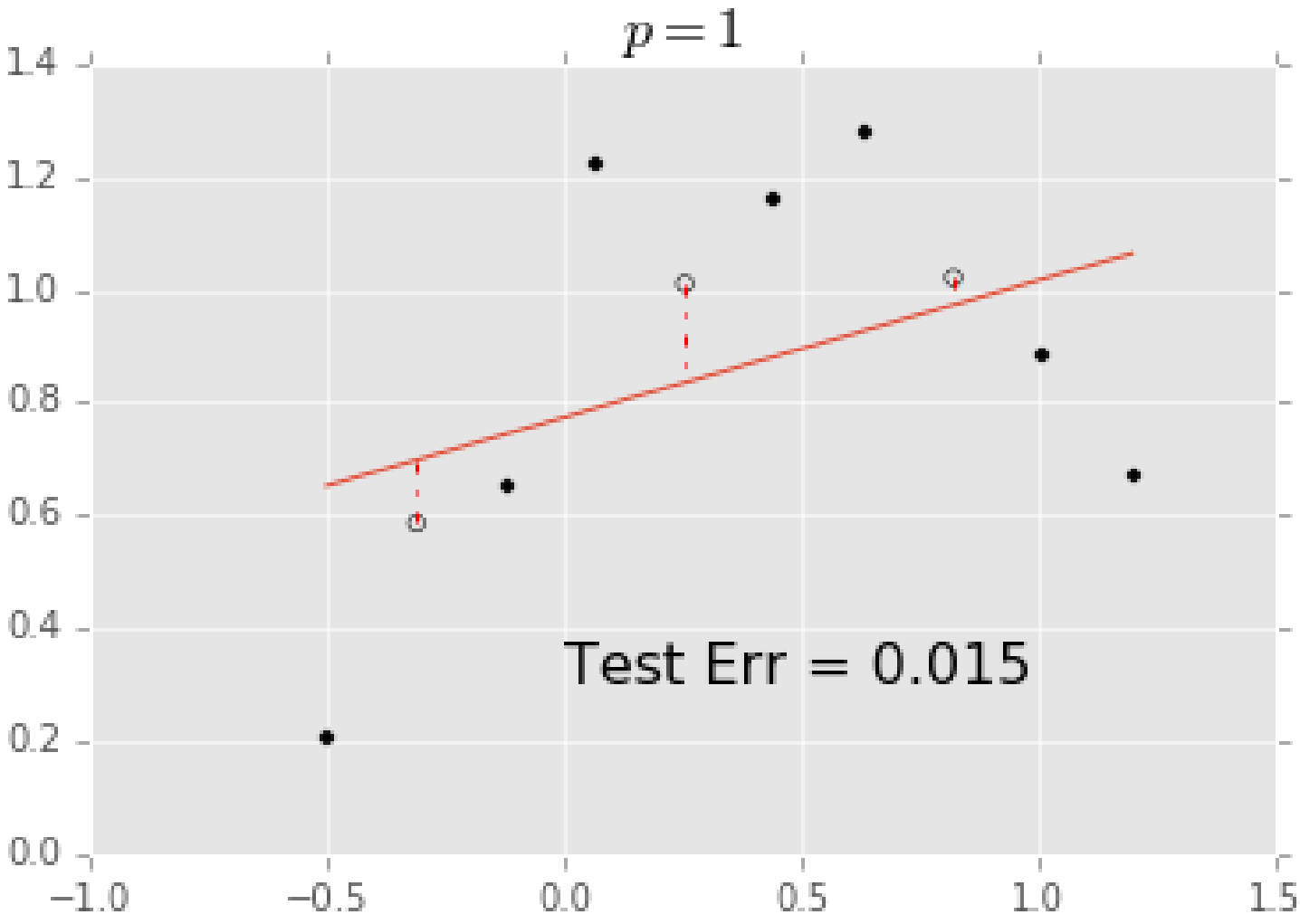
Test Error



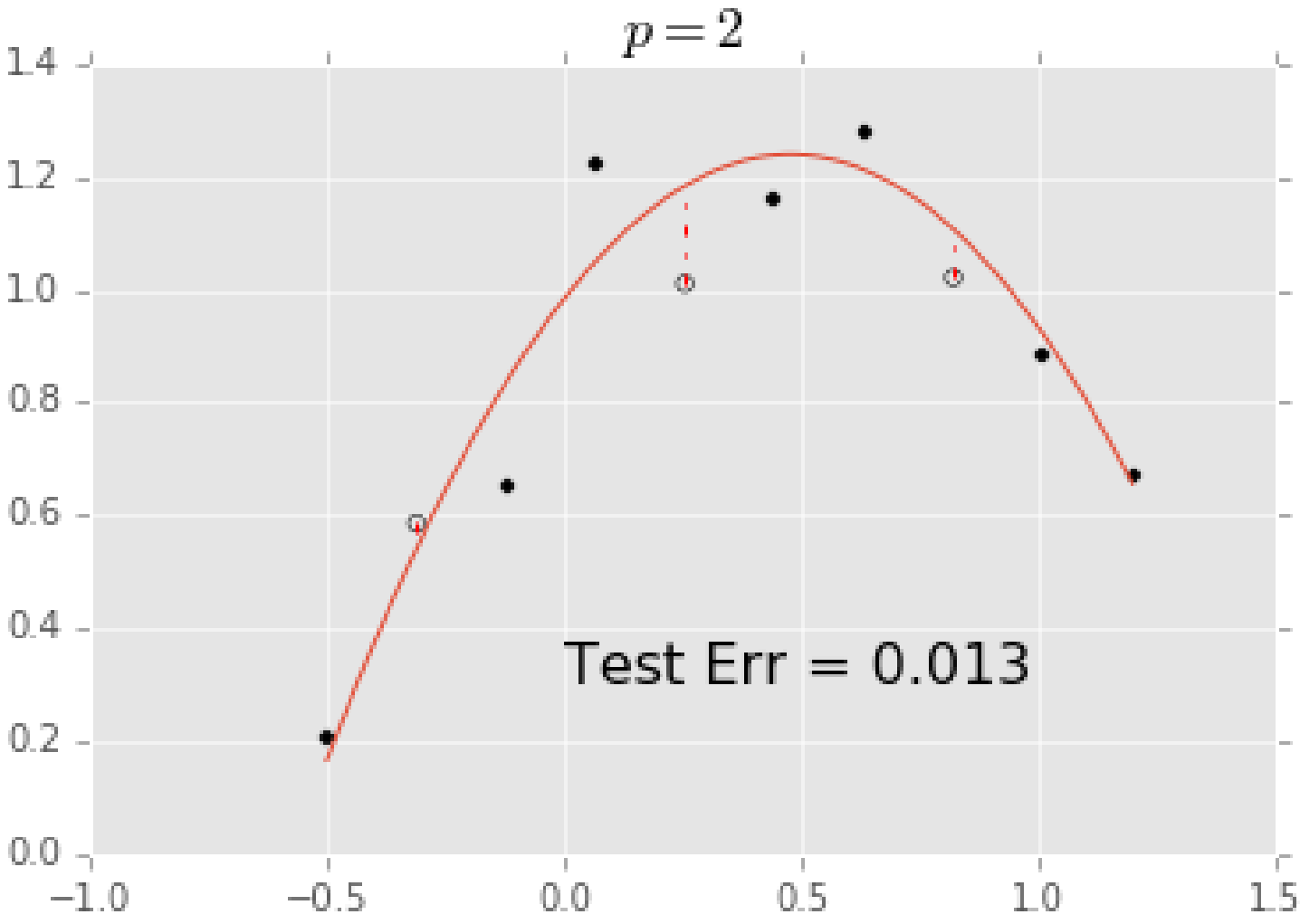
Test Error



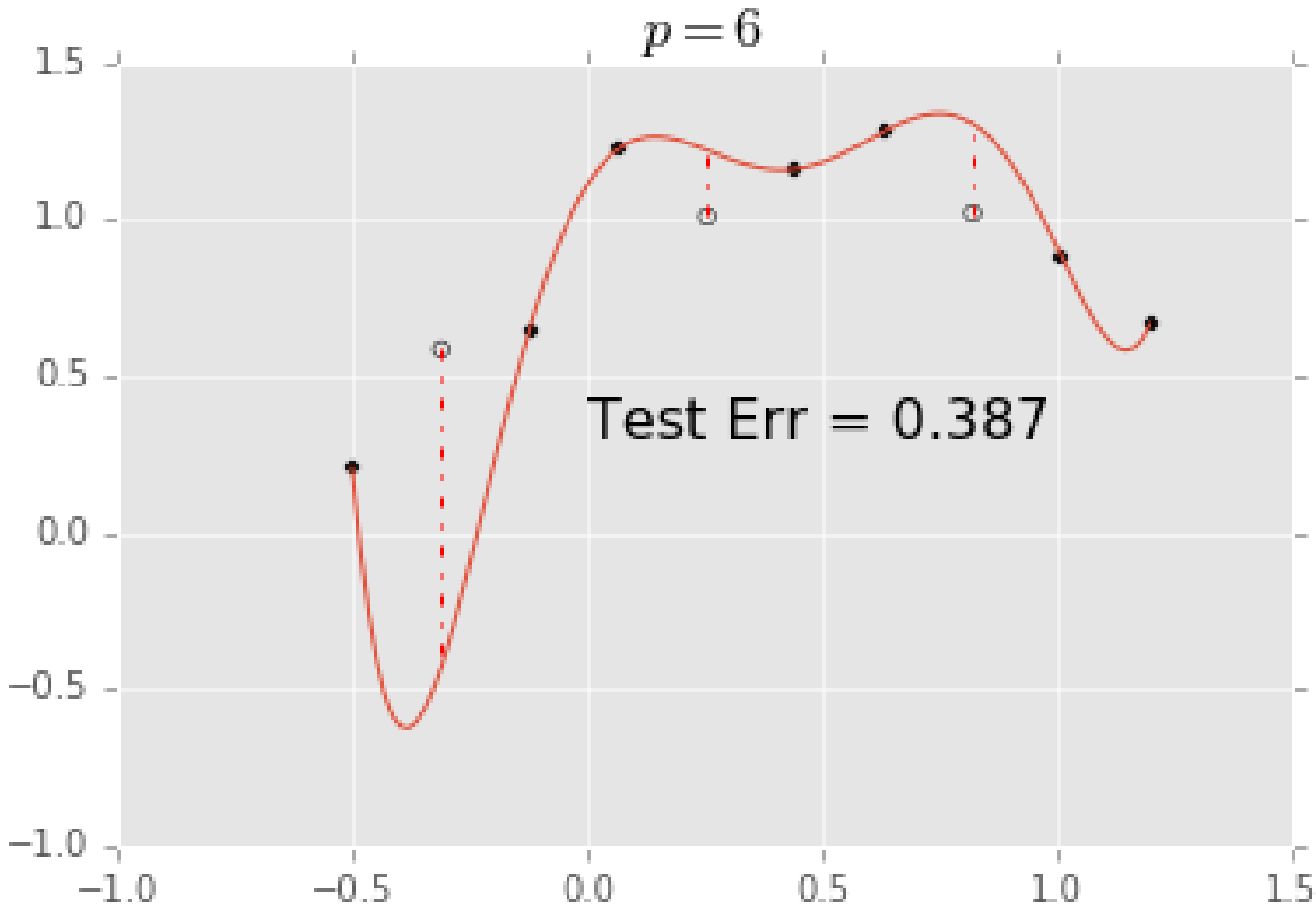
Test Error



Test Error



Test Error



Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.

Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:

Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.

Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.

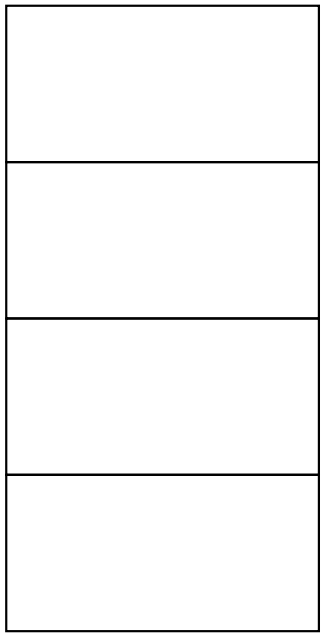
Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.
  3. In the end, we will get k estimates of the prediction error.

Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.
  3. In the end, we will get k estimates of the prediction error.

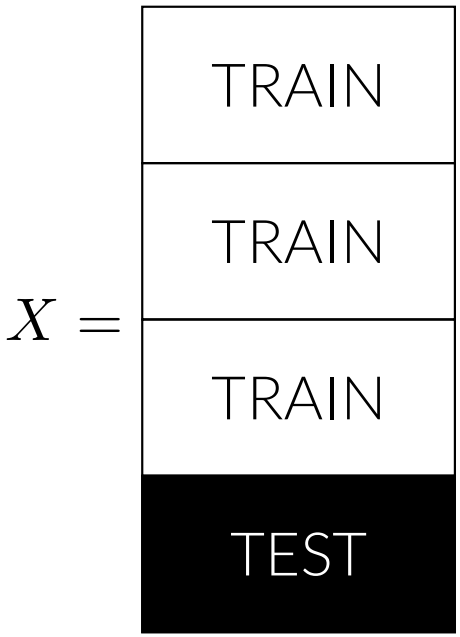
Example of 4-fold cross validation

$X =$ 

Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.
  3. In the end, we will get k estimates of the prediction error.

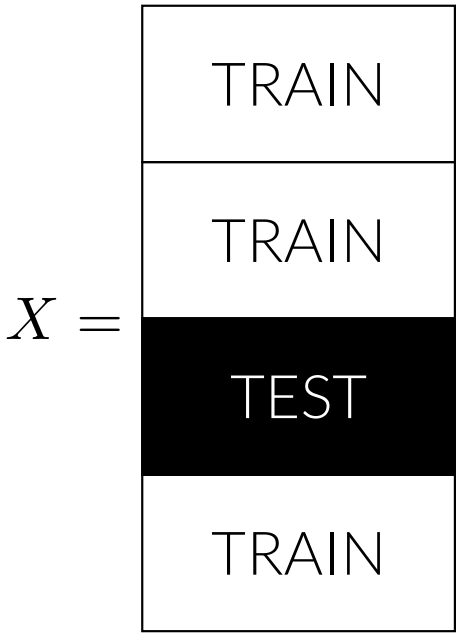
Example of 4-fold cross validation



Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.
  3. In the end, we will get k estimates of the prediction error.

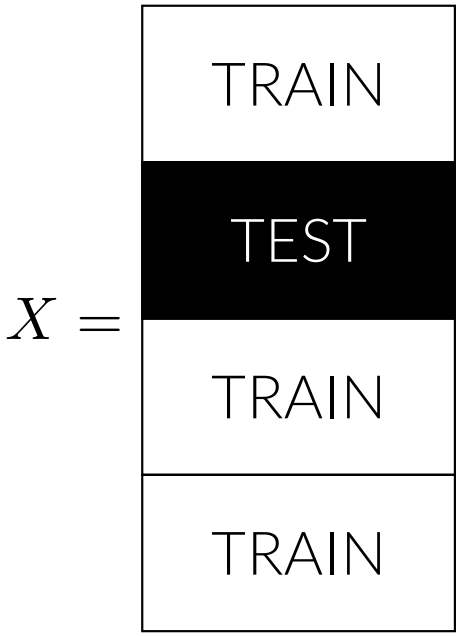
Example of 4-fold cross validation



Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.
  3. In the end, we will get k estimates of the prediction error.

Example of 4-fold cross validation

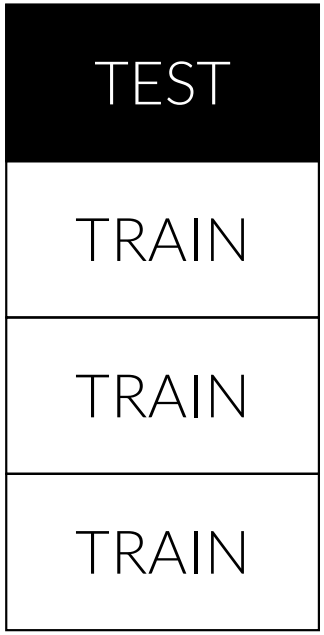


Cross Validation

- You might be concerned that our estimate of the prediction error was based on just one test set of 3 observations.
- Cross validation creates several test sets, as follows:
  1. First, the data is divided into k **folds**.
  2. One at a time, each fold is used as the test set and the model is trained on the remaining data. Then, the test error is calculated using the held-out fold.
  3. In the end, we will get k estimates of the prediction error.

Example of 4-fold cross validation

$X =$



Cross Validation in Scikit Learn

Scikit-learn provides utilities that will do cross validation for you.

```
from sklearn.linear_model import LinearRegression
from sklearn.cross_validation import cross_val_score

model = LinearRegression()
-cross_val_score(model, X, y, cv=5, scoring="mean_squared_error")
```

In-Class Exercise

In-Class Exercise

Use the autos data set. Find the "best" polynomial model of price in terms of city-mpg. You may use AIC, BIC, or cross validation.

In-Class Exercise

In-Class Exercise

Use the autos data set. Find the "best" polynomial model of price in terms of city-mpg. You may use AIC, BIC, or cross validation.

```
model = LinearRegression()
train_errs, test_errs = [], []

for p in range(1, 5):
    # add x^p to the DataFrame
    data["city-mpg^%d" % p] = data["city-mpg"] ** p
    # get the training error
    X = data[["city-mpg^%d" % i in range(1, p)]]
    model.fit(X, y)
    y_hat = model.predict(X)
    train_errs.append( np.mean((y - y_hat) ** 2) )
    # get the test error
    test_errs.append( np.mean(-cross_val_score(
                model, X, y, cv=5,
                scoring="mean_squared_error")) )
```