Ensemble methods: bagging

By Marios Michailidis

Examined ensemble methods

- Averaging (or blending)
- Weighted averaging
- Conditional averaging
- Bagging
- Boosting
- Stacking
- StackNet

What is Bagging

Means **averaging** slightly different versions of the same model to improve accuracy

Why Bagging

- There are 2 main sources of errors in modelling:
    1. Errors due to **Bias** (underfitting)
    2. Errors due **Variance** (overfitting)

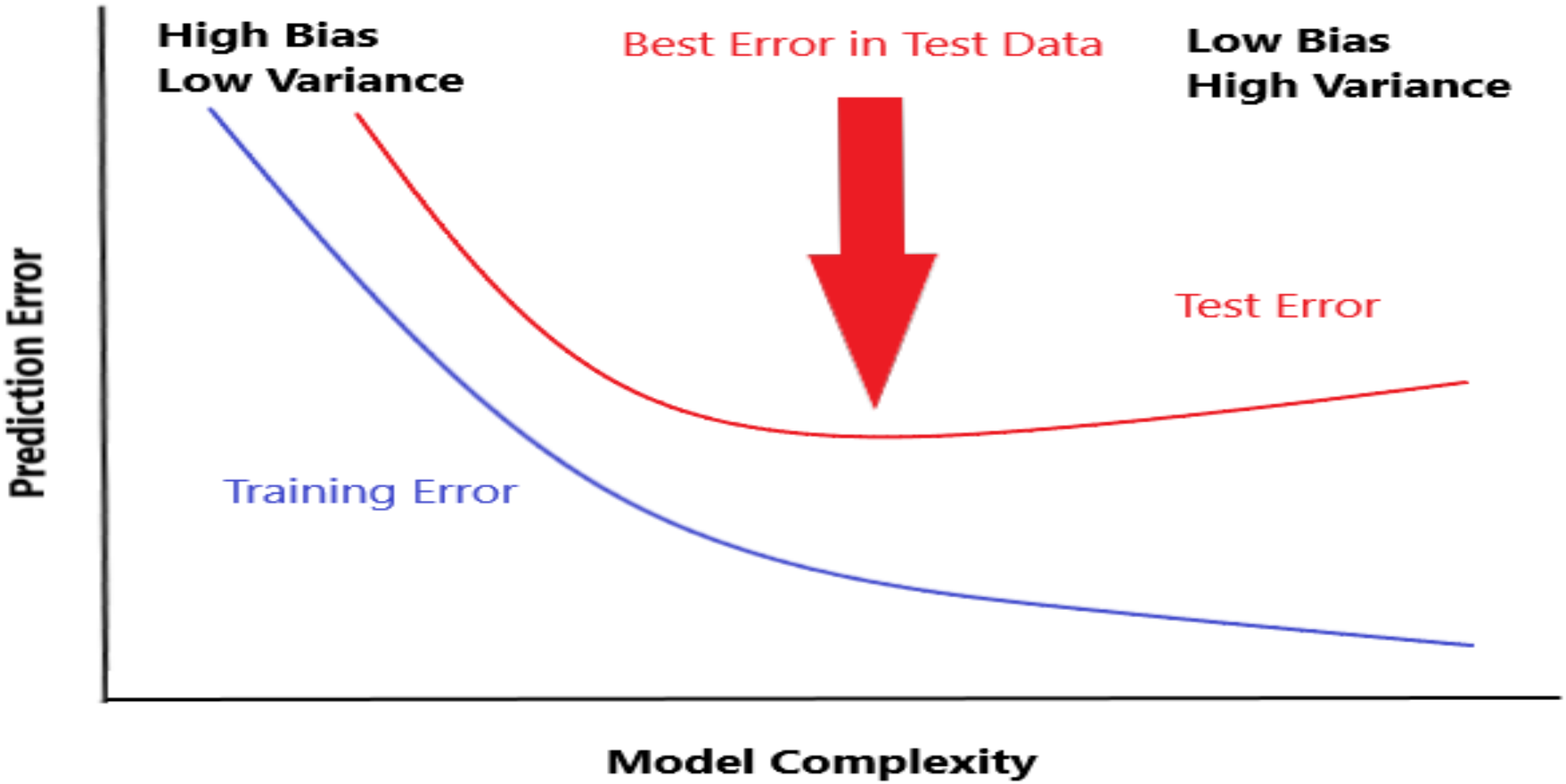
Why Bagging

Why Bagging

Why Bagging

Jon

Why Bagging


High Bias
Low Variance
Best Error in Test Data
Low Bias
High Variance
Test Error
Training Error
Prediction Error
Model Complexity

Parameters that control bagging?

- Changing the seed
- Row (Sub) sampling or Bootstrapping
- Shuffling
- Column (Sub) sampling
- Model-specific parameters
- Number of models (or bags)
- (Optionally) parallelism

Examples of bagging

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
     model.set_params(random_state=seed + n)# update seed
     model.fit(train,y) # fit model
     preds=model.predict(test) # predict on test data
     bagged_prediction+=preds # add predictions to bagged predictions
#take average of predictions
bagged_prediction/= bags
```

Examples of bagging

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
      model.set_params(random_state=seed + n)# update seed
      model.fit(train,y) # fit model
      preds=model.predict(test) # predict on test data
      bagged_prediction+=preds # add predictions to bagged predictions
#take average of predictions
bagged_prediction/= bags
```

Examples of bagging

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
      model.set_params(random_state=seed + n)# update seed
      model.fit(train,y) # fit model
      preds=model.predict(test) # predict on test data
      bagged_prediction+=preds # add predictions to bagged predictions
#take average of predictions
bagged_prediction/= bags
```

Examples of bagging

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
      model.set_params(random_state=seed + n)# update seed
      model.fit(train,y) # fit model
      preds=model.predict(test) # predict on test data
      bagged_prediction+=preds # add predictions to bagged predictions
#take average of predictions
bagged_prediction/= bags
```

Examples of bagging

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
     model.set_params(random_state=seed + n)# update seed
     model.fit(train,y) # fit model
     preds=model.predict(test) # predict on test data
     bagged_prediction+=preds # add predictions to bagged predictions
#take average of predictions
bagged_prediction/= bags
```

Examples of bagging

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
      model.set_params(random_state=seed + n)# update seed
      model.fit(train,y) # fit model
      preds=model.predict(test) # predict on test data
      bagged_prediction+=preds # add predictions to bagged predictions
#take average of predictions
bagged_prediction/= bags
```