Diagnostics for Choosing a Model

Chris Cornwell

April 15, 2025

Outline

Making choices about your model

High Bias vs. High Variance

Cross-Validation

Outline

Making choices about your model

High Bias vs. High Variance

Cross-Validation

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.

Given our type of machine learning model, what are properties that can be “tuned” to change the Bias and Variance?

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.
Given our type of machine learning model, what are properties that can be "tuned" to change the Bias and Variance?

- Initial, but not perfect, answer is the total number of parameters (i.e., the numbers, or *weights*, that are updated by the procedure which, from training data $\mathcal{S}$, determines f_S). Sometimes, more of these parameters = higher Variance, but this is not definitive and depends on other choices and model type.

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.
Given our type of machine learning model, what are properties that can be "tuned" to change the Bias and Variance?

- Initial, but not perfect, answer is the total number of parameters (i.e., the numbers, or *weights*, that are updated by the procedure which, from training data $\mathcal{S}$, determines f_S). Sometimes, more of these parameters = higher Variance, but this is not definitive and depends on other choices and model type.
- After determining the type of model, choices are made before training that affect both the hypothesis class and training procedure. These are **hyperparameters**. Examples of hyperparameters:

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.
Given our type of machine learning model, what are properties that can be "tuned" to change the Bias and Variance?

- Initial, but not perfect, answer is the total number of parameters (i.e., the numbers, or *weights*, that are updated by the procedure which, from training data $\mathcal{S}$, determines f_S). Sometimes, more of these parameters = higher Variance, but this is not definitive and depends on other choices and model type.
- After determining the type of model, choices are made before training that affect both the hypothesis class and training procedure. These are **hyperparameters**. Examples of hyperparameters:
  1. For a decision tree model: the `max_depth` of the tree.

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.
Given our type of machine learning model, what are properties that can be "tuned" to change the Bias and Variance?

- Initial, but not perfect, answer is the total number of parameters (i.e., the numbers, or *weights*, that are updated by the procedure which, from training data $\mathcal{S}$, determines f_S). Sometimes, more of these parameters = higher Variance, but this is not definitive and depends on other choices and model type.
- After determining the type of model, choices are made before training that affect both the hypothesis class and training procedure. These are **hyperparameters**. Examples of hyperparameters:
  1. For a decision tree model: the `max_depth` of the tree.
  2. For a neural network model: the number of `hidden layers` in the network.

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.
Given our type of machine learning model, what are properties that can be "tuned" to change the Bias and Variance?

- Initial, but not perfect, answer is the total number of parameters (i.e., the numbers, or *weights*, that are updated by the procedure which, from training data $\mathcal{S}$, determines f_S). Sometimes, more of these parameters = higher Variance, but this is not definitive and depends on other choices and model type.
- After determining the type of model, choices are made before training that affect both the hypothesis class and training procedure. These are **hyperparameters**. Examples of hyperparameters:
  1. For a decision tree model: the `max_depth` of the tree.
  2. For a neural network model: the number of `hidden layers` in the network.
  3. For a support vector machine: the type of kernel that is used – e.g., if you will use a polynomial kernel and the `degree` of the polynomial kernel.

Intro

Previously, discussed the Bias-Variance trade-off and making a careful choice of hypothesis class of functions that, for your data, does not exhibit too much Bias, nor too much Variance.
Given our type of machine learning model, what are properties that can be "tuned" to change the Bias and Variance?

- Initial, but not perfect, answer is the total number of parameters (i.e., the numbers, or *weights*, that are updated by the procedure which, from training data $\mathcal{S}$, determines f_S). Sometimes, more of these parameters = higher Variance, but this is not definitive and depends on other choices and model type.
- After determining the type of model, choices are made before training that affect both the hypothesis class and training procedure. These are **hyperparameters**. Examples of hyperparameters:
  1. For a decision tree model: the `max_depth` of the tree.
  2. For a neural network model: the number of `hidden layers` in the network.
  3. For a support vector machine: the type of kernel that is used – e.g., if you will use a polynomial kernel and the `degree` of the polynomial kernel.
  4. For clustering: for K-means clustering, the number `K`; for `DBSCAN` clustering, the radius `epsilon`.

Choosing Hyperparameters

How do you choose the hyperparameters?

[1] Don't touch the test data until the model, with parameters, is trained and ready to perform on data.

[2] In particular, if the hyperparameter is real-valued, then you have picked a partition (or step-size) in the interval. For example, $0.0, 0.05, 0.1, \ldots, 0.95, 1.0$ in interval $[0, 1]$.

Choosing Hyperparameters

How do you choose the hyperparameters?
As always, first split your data into training data and test data.[1] Common splits of training and test data are 70/30 percent or 80/20 percent for training/test (not a "hard and fast" rule).

[1] Don't touch the test data until the model, with parameters, is trained and ready to perform on data.

[2] In particular, if the hyperparameter is real-valued, then you have picked a partition (or step-size) in the interval. For example, $0.0, 0.05, 0.1, \ldots, 0.95, 1.0$ in interval $[0, 1]$.

Choosing Hyperparameters

How do you choose the hyperparameters?
As always, first split your data into training data and test data.[1] Common splits of training and test data are 70/30 percent or 80/20 percent for training/test (not a "hard and fast" rule).
One method: a **grid search**. Say that you have M hyperparameters. Further, say that for the j^{th} hyperparameter you have chosen a "reasonable" range of values, such that N_j values of the hyperparameter are possible.[2]

[1] Don't touch the test data until the model, with parameters, is trained and ready to perform on data.

[2] In particular, if the hyperparameter is real-valued, then you have picked a partition (or step-size) in the interval. For example, $0.0, 0.05, 0.1, \ldots, 0.95, 1.0$ in interval $[0, 1]$.

Choosing Hyperparameters

How do you choose the hyperparameters?
As always, first split your data into training data and test data.[1] Common splits of training and test data are 70/30 percent or 80/20 percent for training/test (not a "hard and fast" rule).
One method: a **grid search**. Say that you have M hyperparameters. Further, say that for the j^{th} hyperparameter you have chosen a "reasonable" range of values, such that N_j values of the hyperparameter are possible.[2]
Let $\mathcal{L}_{test}$ be the loss value on the test data. The set of all possible M-tuples of hyperparameters has size $\prod_{j=1}^{M} N_j$. A grid search method is to train a model for each of the possible M-tuples and then compute $\mathcal{L}_{test}$. Then, you choose the model with the smallest $\mathcal{L}_{test}$ value.

[1] Don't touch the test data until the model, with parameters, is trained and ready to perform on data.

[2] In particular, if the hyperparameter is real-valued, then you have picked a partition (or step-size) in the interval. For example, $0.0, 0.05, 0.1, \ldots, 0.95, 1.0$ in interval $[0, 1]$.

Choosing Hyperparameters

How do you choose the hyperparameters?
As always, first split your data into training data and test data.[1] Common splits of training and test data are 70/30 percent or 80/20 percent for training/test (not a "hard and fast" rule).
One method: a **grid search**. Say that you have M hyperparameters. Further, say that for the j^{th} hyperparameter you have chosen a "reasonable" range of values, such that N_j values of the hyperparameter are possible.[2]
Let $\mathcal{L}_{test}$ be the loss value on the test data. The set of all possible M-tuples of hyperparameters has size $\prod_{j=1}^{M} N_j$. A grid search method is to train a model for each of the possible M-tuples and then compute $\mathcal{L}_{test}$. Then, you choose the model with the smallest $\mathcal{L}_{test}$ value.
Downsides: most notably, as described it *uses the test data* to make a decision about the parameters. Can potentially cause overfitting.
It is also typically computationally expensive.

---

[1] Don't touch the test data until the model, with parameters, is trained and ready to perform on data.

[2] In particular, if the hyperparameter is real-valued, then you have picked a partition (or step-size) in the interval. For example, $0.0, 0.05, 0.1, \ldots, 0.95, 1.0$ in interval $[0, 1]$.

Validation Sets

In addition to splitting the data into a training set and test set, it is common practice to have an additional split of the training set, with some amount put into a **validation set**.

Validation Sets

In addition to splitting the data into a training set and test set, it is common practice to have an additional split of the training set, with some amount put into a **validation set**.
The validation data may then be used to assess how good the hyperparameter choices are – computing the loss on validation data, $\mathcal{L}_{valid}$ say – and we are able to keep the test data locked away until the end, for assessment of the final model.

Validation Sets

In addition to splitting the data into a training set and test set, it is common practice to have an additional split of the training set, with some amount put into a **validation set**.

The validation data may then be used to assess how good the hyperparameter choices are – computing the loss on validation data, $\mathcal{L}_{valid}$ say – and we are able to keep the test data locked away until the end, for assessment of the final model.

For example, a grid search method might be applied, but choosing the model with smallest $\mathcal{L}_{valid}$, rather than $\mathcal{L}_{test}$, and this will help the $\mathcal{L}_{test}$ of the final model be a better estimate of the expected population loss.

Outline

Making choices about your model

High Bias vs. High Variance

Cross-Validation

Diagnostic to Test for High Bias

Use validation data to diagnose if chosen hyperparameters create too much Bias or too much Variance.

Diagnostic to Test for High Bias

Use validation data to diagnose if chosen hyperparameters create too much Bias or too much Variance.

- In training/validation/test split of data, say that n is the number of points in training subset. For a series of values of m with $m \leq n$, train a model with your choice of hyperparameters, with just m of points from training data. Then, compute the value of $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ on each of the resulting models.

Diagnostic to Test for High Bias

Use validation data to diagnose if chosen hyperparameters create too much Bias or too much Variance.

- In training/validation/test split of data, say that n is the number of points in training subset. For a series of values of m with $m \leq n$, train a model with your choice of hyperparameters, with just m of points from training data. Then, compute the value of $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ on each of the resulting models.
- Plot the curves for $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$, as functions of m. In a scenario with high Bias, they appear as depicted below. Note that the $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ curves approach each other as m increases; however, the loss (even on training data) is too high.

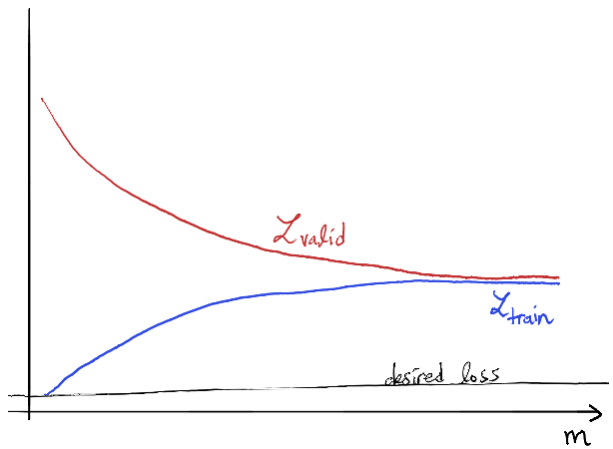


Figure: High Bias Scenario; m is the # of data points used in training.

Diagnostic to Test for High Variance

- However, if $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ are computed as before, and their curves plotted as functions of m, then in a scenario with high Variance, they will appear as depicted below. Here, between the $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ curves there is a gap, that remains as m increases.

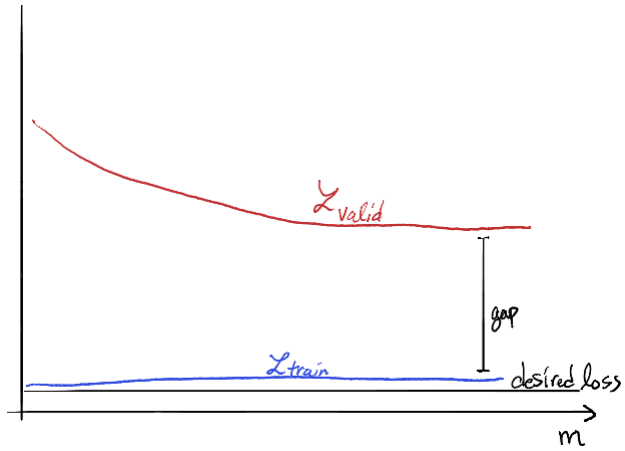


Figure: High Variance Scenario; m is the # of data points used in training.

Diagnostic to Test for High Variance

- However, if $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ are computed as before, and their curves plotted as functions of m, then in a scenario with high Variance, they will appear as depicted below. Here, between the $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ curves there is a gap, that remains as m increases.

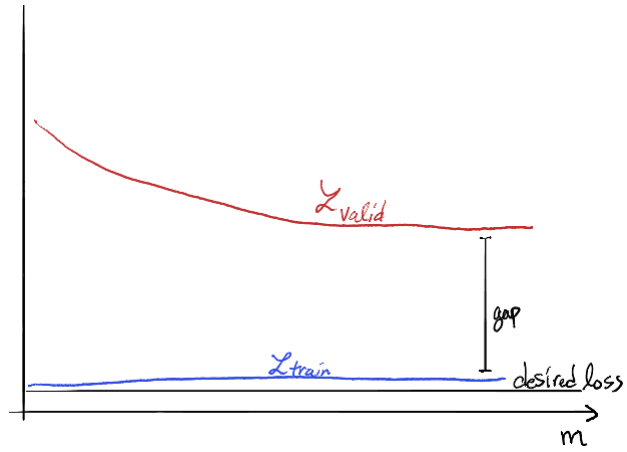


Figure: High Variance Scenario; m is the # of data points used in training.

In practice, the curves are not so smooth; e.g., the order in which training points are added (as m increases) to the training set has an effect on getting low training (or validation) loss score.

Diagnostic to Test for High Variance

- However, if $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ are computed as before, and their curves plotted as functions of m, then in a scenario with high Variance, they will appear as depicted below. Here, between the $\mathcal{L}_{train}$ and $\mathcal{L}_{valid}$ curves there is a gap, that remains as m increases.

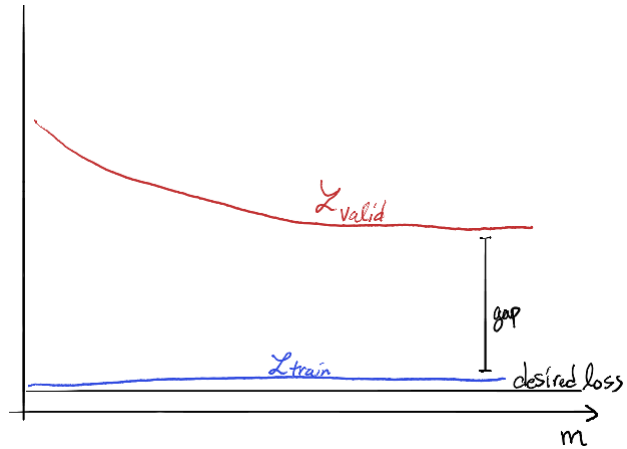


Figure: High Variance Scenario; m is the # of data points used in training.

In practice, the curves are not so smooth; e.g., the order in which training points are added (as m increases) to the training set has an effect on getting low training (or validation) loss score.

- To see the above shape, may need to randomly re-order the training data multiple times, redo the computations, and plot the averages as the curve.

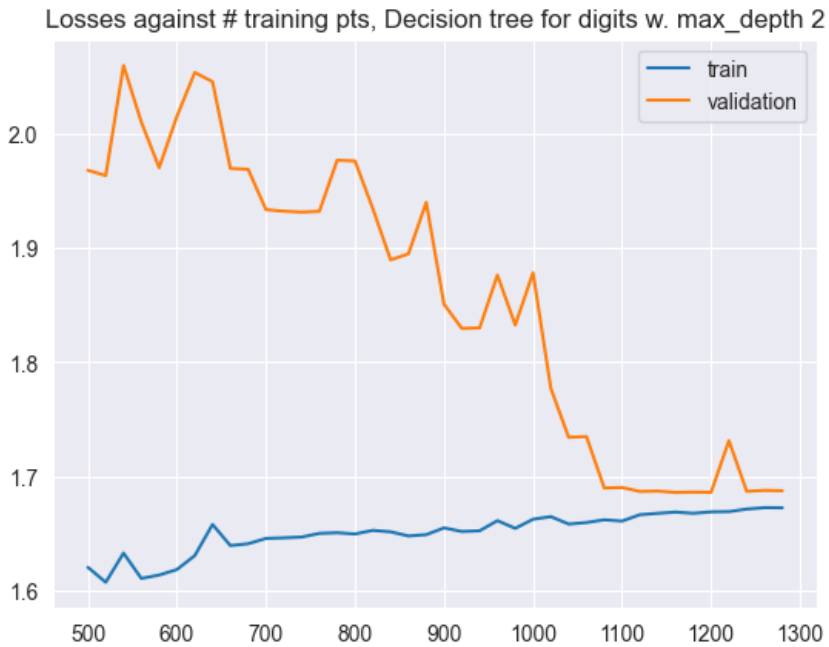
Diagnostic for high Bias versus high Variance

Recall from last lecture, the decision tree (with no maximum depth) which was fit to classify the digit in a handwritten image.
Below, the diagnostic was run when setting the maximum depth of the tree equal to 2. The curve is indicative of the Bias being high. The number of points in the training set is along the horizontal axis. [3]

[3] Remark: for this data, the log loss value shown for training data meant that less than 50% of the images were being classified correctly.

Diagnostic for high Bias versus high Variance

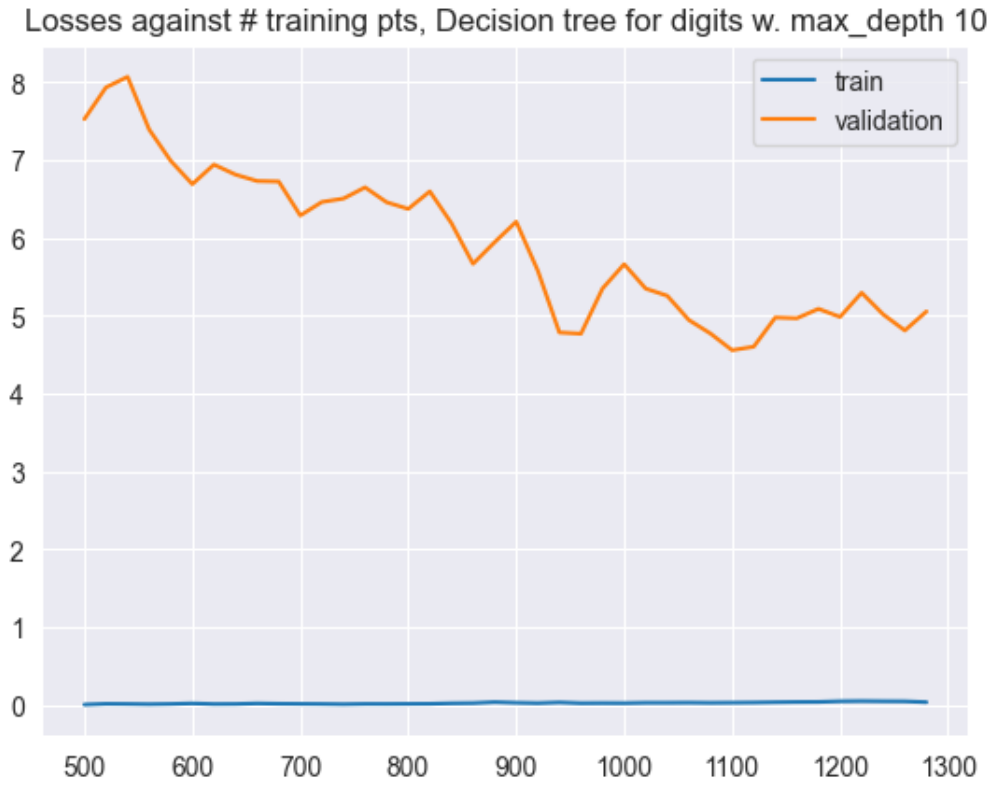
Recall from last lecture, the decision tree (with no maximum depth) which was fit to classify the digit in a handwritten image.
Below, the diagnostic was run when setting the maximum depth of the tree equal to 2. The curve is indicative of the Bias being high. The number of points in the training set is along the horizontal axis. [3]



[3] Remark: for this data, the log loss value shown for training data meant that less than 50% of the images were being classified correctly.

Diagnostic for high Bias versus high Variance

The same diagnostic was run, but setting the maximum depth of the tree equal to 10. The curve is shown below and is indicative of a scenario when Variance is high.



Outline

Cross-Validation

If you are not in a "data rich" setting – that is, you do not have a large amount of data – then setting aside 10-20% of it as validation data might make it difficult to train a model that performs well. Additionally, *which* data goes into the validation subset might affect the resulting prediction function, and its performance, significantly.

A remedy is to use k-fold **cross-validation**. Commonly, practitioners use 5-fold or 10-fold cross-validation. Here, we describe the 5-fold version.

After having separated out the test data, randomly sort your remaining data into 5 subsets. For example, if `x` is the array to be separated into 5 arrays, the following would do the job.

```
indices = np.arange(len(x))
np.random.shuffle(indices)
n_subset = int(len(x)/5)
for i in range(5):
    subsets[i] = x[i*n_subset:(i+1)*n_subset]
```

Next, you train and do validation on 5 models, with training sets determined from the subsets:

train	train	train	train	valid
train	train	train	valid	train
train	train	valid	train	train
train	valid	train	train	train
valid	train	train	train	train