Lecture 1: Course information, supervised vs. unsupervised learning, bias-variance tradeoff

Reading: Chapter 2

GU4241/GR5241 Statistical Machine Learning

Linxi Liu

January 17, 2017

Course Information

- All resources from class will be posted on Canvas https://courseworks.columbia.edu/welcome/. Check the web site often for any important course-related announcements.

Course Information

- All resources from class will be posted on Canvas https://courseworks.columbia.edu/welcome/. Check the web site often for any important course-related announcements.
- We will mainly use R and Python for data analysis.

Course Information

- All resources from class will be posted on Canvas https://courseworks.columbia.edu/welcome/. Check the web site often for any important course-related announcements.

- We will mainly use R and Python for data analysis.

- **Homeworks:**
  a. There will be six assignments. See course schedule for detailed information.

Course Information

- All resources from class will be posted on Canvas https://courseworks.columbia.edu/welcome/. Check the web site often for any important course-related announcements.

- We will mainly use R and Python for data analysis.

- **Homeworks:**
  a. There will be six assignments. See course schedule for detailed information.

  b. We **DO NOT** accept late homework.

Course Information

- All resources from class will be posted on Canvas https://courseworks.columbia.edu/welcome/. Check the web site often for any important course-related announcements.
- We will mainly use R and Python for data analysis.
- **Homeworks:**
  a. There will be six assignments. See course schedule for detailed information.
  b. We **DO NOT** accept late homework.
  c. The lowest score will be dropped.

Textbook

Textbook
T. Hastie, R. Tibshirani and J. Friedman. *The Elements of Statistical Learning.* Second Edition, Springer, 2009

Textbook

Textbook
T. Hastie, R. Tibshirani and J. Friedman. *The Elements of Statistical Learning.* Second Edition, Springer, 2009

References
G. James, D. Witten, T. Hastie and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R.* Springer, 2013

K. P. Murphy. *Machine Learning: a Probabilistic Perspective.* MIT Press, 2012.

J. Shawe-Taylor and N. Cristianini. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, 2000.

D. Barber. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012.

Grading

Your overall course grade will be determined as a weighted average of the following categories:

40 %	homework assignments
25 %	midterm exam
35 %	final exam

Exams

- **Midterm**
  Thu, March 9, 2017, 6:10pm - 7:25pm, *in lecture*

Exams

- **Midterm**
  Thu, March 9, 2017, 6:10pm - 7:25pm, *in lecture*

- **Final**
  Tue, May 9, 2017, 7:10pm - 10pm *location TBA*

Exams

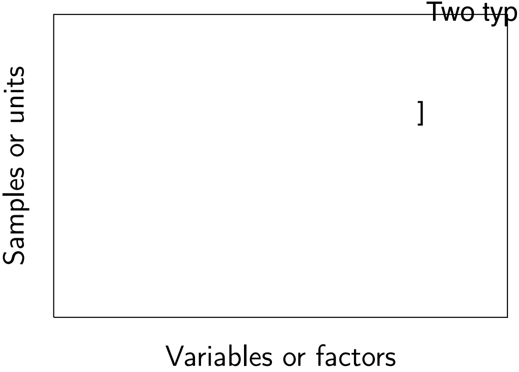
- **Midterm**
  Thu, March 9, 2017, 6:10pm - 7:25pm, *in lecture*

- **Final**
  Text
  Tue, May 9, 2017, 7:10pm - 10pm *location TBA*

- In general, **NO MAKE-UP EXAMES** are granted. If an emergency occurs on the exam day, you must contact the instructor before the exam.

Supervised vs. unsupervised learning

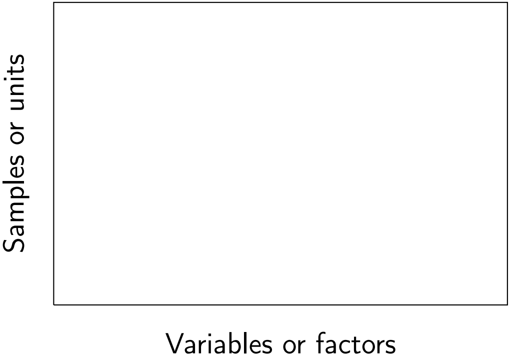
In **unsupervised learning** we start with a data matrix:



Two types of the machine learning

Supervised vs. unsupervised learning

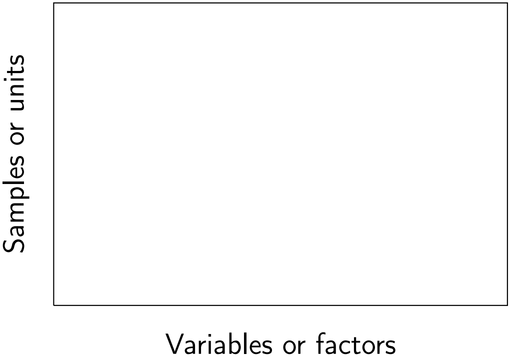
In **unsupervised learning** we start with a data matrix:



Quantitative, eg. weight, height, number of children, ...

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:



Qualitative, eg. college major, profession, gender, ...

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

Our goal is to:

- Find meaningful relationships between the variables or units.

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

Our goal is to:

- Find meaningful relationships between the variables or units.
- Find low-dimensional representations of the data which make it easy to visualize the variables and units.

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

Our goal is to:

- Find meaningful relationships between the variables or units.
- Find low-dimensional representations of the data which make it easy to visualize the variables and units.
- Find meaningful groupings of the data.

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

Our goal is to:

- Find meaningful relationships between the variables or units. Correlation analysis.
- Find low-dimensional representations of the data which make it easy to visualize the variables and units. PCA, ICA, multidimensional scaling, locally linear embeddings, etc.
- Find meaningful groupings of the data. Clustering.

Supervised vs. unsupervised learning

In **unsupervised learning** we start with a data matrix:

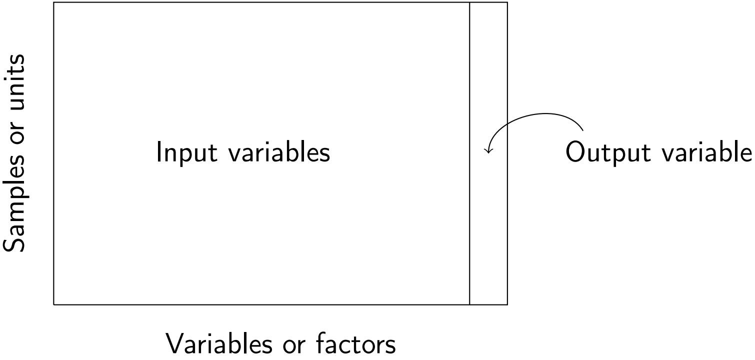
Our goal is to:

- Find meaningful relationships between the variables or units.
- Find low-dimensional representations of the data which make it easy to visualize the variables and units.
- Find meaningful groupings of the data.

Unsupervised learning is also known in Statistics as **exploratory data analysis**.

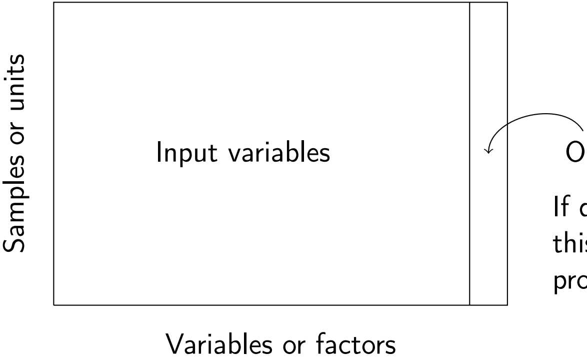
Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:



Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:



Samples or units

Input variables

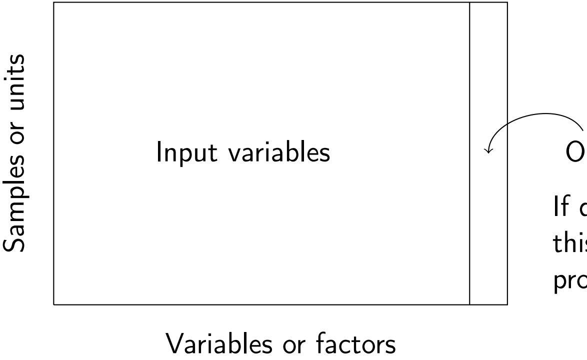
Variables or factors

Output variable

If quantitative, we say this is a regression problem

Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:



Output variable

If qualitative, we say this is a classification problem

Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:

If X is the vector of inputs for a particular sample. The output variable is modeled by:

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Supervised vs. unsupervised learning

In **supervised learning**, there are *input* variables, and *output* variables:

If X is the vector of inputs for a particular sample. The output variable is modeled by:

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Our goal is to learn the function f, using a set of training samples.

Supervised vs. unsupervised learning

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Motivations:

- **Prediction:** Useful when the input variable is readily available, but the output variable is not.

  Example: Predict stock prices next week using data from last month.

Supervised vs. unsupervised learning

$$Y = f(X) + \underbrace{\varepsilon}_{\text{Random error}}$$

Motivations:

- **Prediction:** Useful when the input variable is readily available, but the output variable is not.
- **Inference:** A model for f can help us understand the structure of the data — which variables influence the output, and which don't? What is the relationship between each variable and the output, e.g. linear, non-linear?

  Example: What is the influence of genetic variations on the incidence of heart disease.

Parametric and nonparametric methods:

There are two kinds of supervised learning method:

Gausian distribution; mean and variance
Normal distribution:

Parametric and nonparametric methods:

we do not know the specific model parameters assumption, it can be very flexible

There are two kinds of supervised learning method:

- **Parametric methods:** We assume that f takes a specific form. For example, a linear form:

$$f(X) = X_1\beta_1 + \cdots + X_p\beta_p$$

with parameters $\beta_1, \ldots, \beta_p$. Using the training data, we try to *fit* the parameters.

Parametric and nonparametric methods:

There are two kinds of supervised learning method:

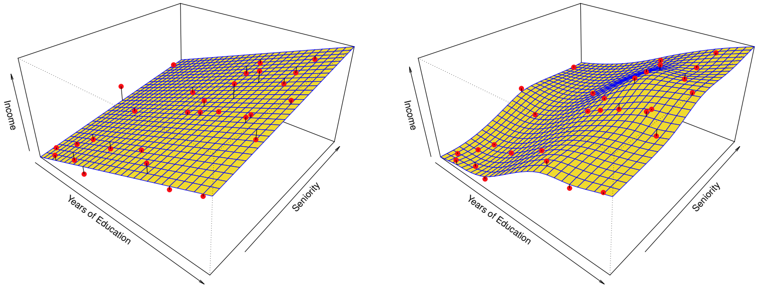
- **Parametric methods:** We assume that f takes a specific form. For example, a linear form:

  $$f(X) = X_1\beta_1 + \cdots + X_p\beta_p$$

  with parameters $\beta_1, \dots, \beta_p$. Using the training data, we try to *fit* the parameters.

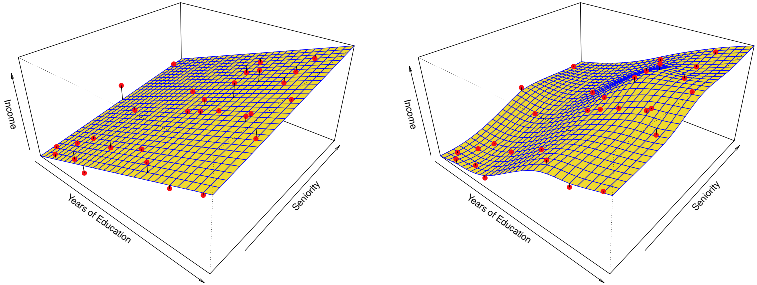
- **Non-parametric methods:** We don't make any assumptions on the form of f, but we restrict how "wiggly" or "rough" the function can be.

Parametric vs. nonparametric prediction



ISL Figures 2.4 and 2.5

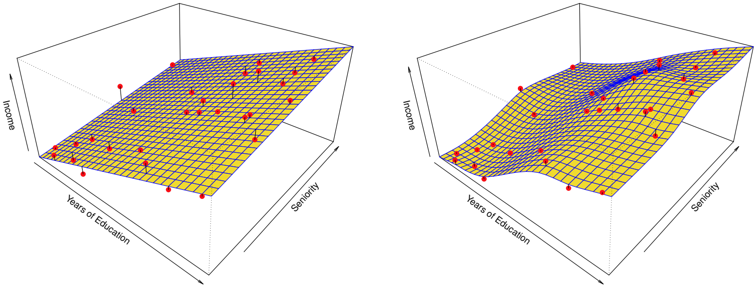
Parametric vs. nonparametric prediction



ISL Figures 2.4 and 2.5

Parametric methods have a limit of fit quality. Non-parametric methods keep improving as we add more data to fit.

Parametric vs. nonparametric prediction



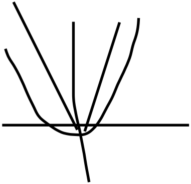
ISL Figures 2.4 and 2.5

Parametric methods have a limit of fit quality. Non-parametric methods keep improving as we add more data to fit.

Parametric methods are often simpler to interpret.

Loss Function

The loss function $L(Y, \hat{f}(X))$ measures the errors between the observed value Y and the predicted value $\hat{f}(X)$.



Loss Function

The loss function $L(Y, \hat{f}(X)$ measures the errors between the observed value Y and the predicted value $\hat{f}(X)$.

In a regression problem, two most common loss functions are:

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ |Y - \hat{f}(X)| & \text{absolute error.} \end{cases}$$

absolute error is more sensitive, and produce more penalty

Prediction error

Training data: $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$ to get a model
Predicted function: $\hat{f}$.

Our goal in supervised learning is to minimize the expected prediction error.

Prediction error

Training data: $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$
Predicted function: $\hat{f}$.

Our goal in supervised learning is to minimize the expected prediction error. Under squared-error loss, this is the *Mean Squared Error*:

$$MSE(\hat{f}) = E(y_0 - \hat{f}(x_0))^2.$$

randomness from train data set, and the test data set

Prediction error

Training data: $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$
Predicted function: $\hat{f}$.

Our goal in supervised learning is to minimize the expected prediction error. Under squared-error loss, this is the *Mean Squared Error*:

$$MSE(\hat{f}) = E(y_0 - \hat{f}(x_0))^2.$$

Unfortunately, this quantity cannot be computed, because we don't know the joint distribution of (X, Y).

Prediction error

Training data: $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$
Predicted function: $\hat{f}$.

Our goal in supervised learning is to minimize the expected prediction error. Under squared-error loss, this is the *Mean Squared Error*:

$$MSE(\hat{f}) = E(y_0 - \hat{f}(x_0))^2.$$

Unfortunately, this quantity cannot be computed, because we don't know the joint distribution of (X, Y). We can compute a sample average using the training data; this is known as the training MSE:

$$MSE_{\text{training}}(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2.$$

Prediction error

The main challenge of statistical learning is that *a low training MSE does not imply a low MSE.*

Prediction error

The main challenge of statistical learning is that *a low training MSE does not imply a low MSE.*

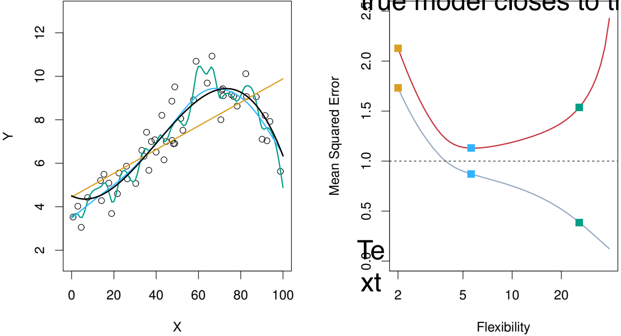
If we have test data $\{(x_i', y_i'); i = 1, \ldots, m\}$ which were not used to fit the model, a better measure of quality for $\hat{f}$ is the test MSE:

$$MSE_{\mathsf{test}}(\hat{f}) = \frac{1}{m} \sum_{i=1}^{m} (y_i' - \hat{f}(x_i'))^2.$$

ISL Figure 2.9.

low: training mse
upper: test mse
true model closes to the linear one

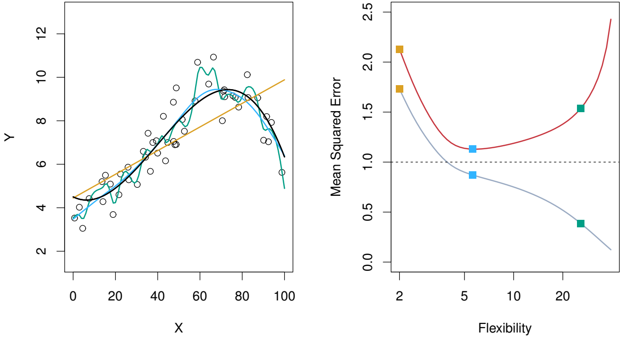
Y; linear regression



Te
xt

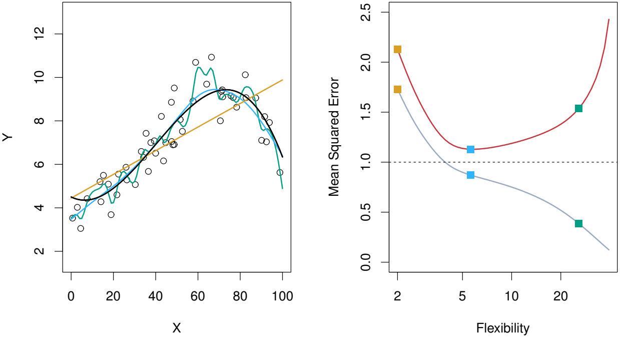
The circles are simulated data from the black curve.

ISL Figure 2.9.



The circles are simulated data from the black curve.
In this artificial example, we *know* what f is.

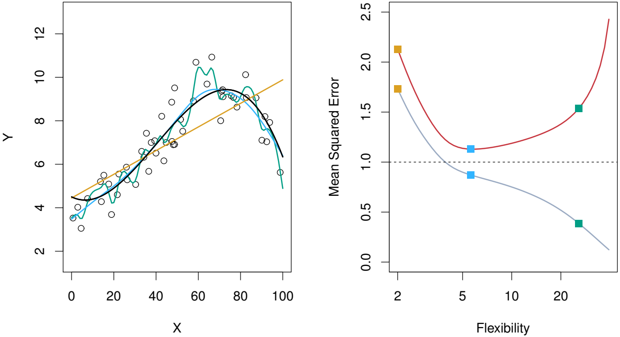
ISL Figure 2.9.



Three estimates $\hat{f}$ are shown:

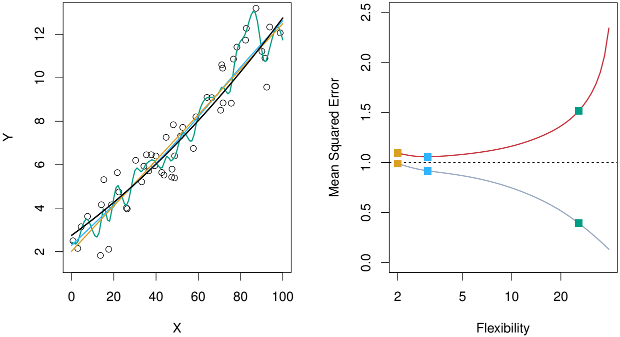
1. Linear regression.
2. Splines (very smooth).
3. Splines (quite rough).

ISL Figure 2.9.



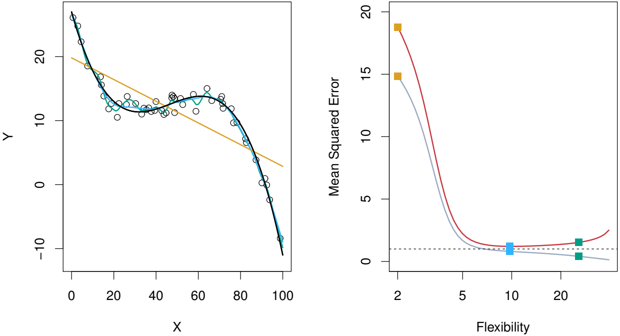
Red line: Test MSE.
Gray line: Training MSE.

ISL Figure 2.10



The function f is now almost linear.

ISL Figure 2.11



When the noise ε has small variance, the third method does well.

only the test mse that we are interested in

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathsf{Var}(\hat{f}(x_0)) + [\mathsf{Bias}(\hat{f}(x_0))]^2 + \mathsf{Var}(\varepsilon_0).$$

first part: we could not reduce

.

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\varepsilon_0).$$

Irreducible error

The bias: how much bias we have from the true model

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon_0).$$

The variance of the estimate of Y: $E[\hat{f}(x_0) - E(\hat{f}(x_0))]^2$

This measures how much the estimate of $\hat{f}$ at x_0 changes when we sample new training data.

The bias variance decomposition

Let x_0 be a fixed test point, $y_0 = f(x_0) + \varepsilon_0$, and $\hat{f}$ be estimated from n training samples $(x_1, y_1) \ldots (x_n, y_n)$.

Let E denote the expectation over y_0 and the training outputs $(y_1, \ldots, y_n)$. Then, the Mean Squared Error at x_0 can be decomposed:

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathsf{Var}(\hat{f}(x_0)) + [\mathsf{Bias}(\hat{f}(x_0))]^2 + \mathsf{Var}(\varepsilon_0).$$

The squared bias of the estimate of Y: $[E(\hat{f}(x_0)) - f(x_0)]^2$

This measures the deviation of the average prediction $\hat{f}(x_0)$ from the truth $f(x_0)$.

Implications of bias variance decomposition

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathsf{Var}(\hat{f}(x_0)) + [\mathsf{Bias}(\hat{f}(x_0))]^2 + \mathsf{Var}(\varepsilon).$$

- The MSE is always positive.

Implications of bias variance decomposition

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathsf{Var}(\hat{f}(x_0)) + [\mathsf{Bias}(\hat{f}(x_0))]^2 + \mathsf{Var}(\varepsilon).$$

- The MSE is always positive.
- Each element on the right hand side is always positive.

Implications of bias variance decomposition

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathsf{Var}(\hat{f}(x_0)) + [\mathsf{Bias}(\hat{f}(x_0))]^2 + \mathsf{Var}(\varepsilon).$$

- The MSE is always positive.
- Each element on the right hand side is always positive.
- Therefore, typically when we decrease the bias beyond some point, we increase the variance, and vice-versa.

we need to make a trid of between the bias and variance;

for non-parametric: it might has high variance

for simple linear regression: the bias could be very high, and the variance could be low

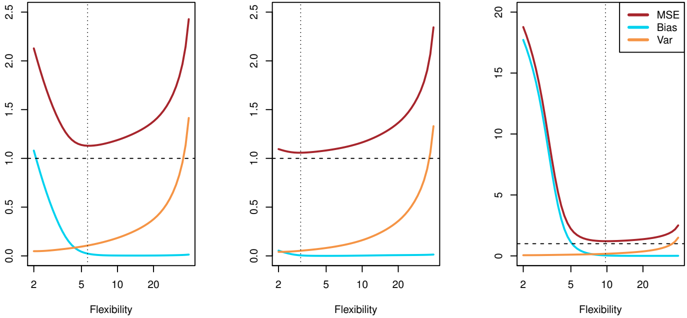
Implications of bias variance decomposition

$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \mathsf{Var}(\hat{f}(x_0)) + [\mathsf{Bias}(\hat{f}(x_0))]^2 + \mathsf{Var}(\varepsilon).$$

- The MSE is always positive.
- Each element on the right hand side is always positive.
- Therefore, typically when we decrease the bias beyond some point, we increase the variance, and vice-versa.

More flexibility $\iff$ Higher variance $\iff$ Lower bias.

Squiggly f, high noise   Linear f, high noise   Squiggly f, low noise



ISL Figure 2.12

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set $\{$Ford, Toyota, Mercedes-Benz, $\ldots\}$.

Text

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set $\{$Ford, Toyota, Mercedes-Benz$, \ldots\}$.

The model:

$$Y = f(X) + \varepsilon$$

becomes insufficient, as f is not necessarily real-valued.

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set $\{$Ford, Toyota, Mercedes-Benz$, \ldots\}$.

The model:

$$\cancel{Y = f(X) + \varepsilon}$$

becomes insufficient, as f is not necessarily real-valued.

Classification problems

In a classification setting, the output takes values in a discrete set.

For example, if we are predicting the brand of a car based on a number of variables, the function f takes values in the set $\{\text{Ford, Toyota, Mercedes-Benz}, \ldots\}$.

We will use slightly different notation:

$$\begin{aligned} P(X, Y) &: \text{joint distribution of } (X, Y), \\ P(Y \mid X) &: \text{conditional distribution of } X \text{ given } Y, \\ \hat{y}_i &: \text{prediction for } x_i. \end{aligned}$$

Loss function for classification

There are many ways to measure the error of a classification prediction. One of the most common is the 0-1 loss:

$$E(\mathbf{1}(y_0 \neq \hat{y}_0))$$

Loss function for classification

There are many ways to measure the error of a classification prediction. One of the most common is the 0-1 loss:

$$E(\mathbf{1}(y_0 \neq \hat{y}_0))$$

Like the MSE, this quantity can be estimated from training and test data by taking a sample average:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(y_i \neq \hat{y}_i)$$

Bayes classifier

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 2

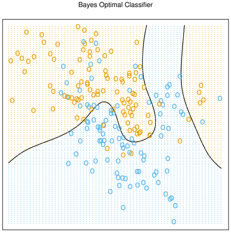


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

In practice, we never know the joint probability P. However, we can assume that it exists.

Bayes is a framework.
P(x=k)
P(x=kly=k)
based on the above two assupmtion,
we can calculate the posterior distribution

Bayes classifier

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 2

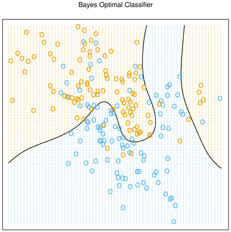


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

In practice, we never know the joint probability P. However, we can assume that it exists.

The Bayes classifier assigns:

$$\hat{y}_i = \text{argmax}_k \;\; P(Y = k \mid X = x_i)$$

It can be shown that this is the best classifier under the 0-1 loss.

Thanks to Sergio Bacallado and Peter Orbanz
for sharing the slides.