

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

Reading: Chapter 2

GU4241/GR5241 Statistical Machine Learning

Linxi Liu  
Jan 19, 2018

## Course Information

- ▶ **Please read the syllabus carefully.**
- ▶ 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.
- ▶ Lecture meets once every week, from 2:40pm to 5:25pm on Friday.
- ▶ Separate labs/review sessions during the week. Students are **REQUIERED** to attend **ONE** lab/review session every week.
- ▶ In each section, students with last name initial from A to L are assigned to the **FIRST** lecture in the week, and those with last name initial from M to Z are assigned to the **SECOND** one.

## Course Mailing List

We have a course mailing list:

[gr5241\\_gu4241\\_course\\_staff \[at\] columbia \[dot\] edu](mailto:gr5241_gu4241_course_staff@columbia.edu)

For any course-related inquiries, please send them to the mailing list.

Please DO NOT email the instructor or the TAs in person. Any email directly sent to the instructor or the TAs WILL NOT get replied.

# Homework

- ▶ We will mainly use R and Python for data analysis.
- ▶ **Homeworks:**
  - a. There will be five 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

## 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:

<b>5%</b>	attendance
<b>35 %</b>	homework assignments
<b>25 %</b>	midterm exam
<b>35 %</b>	final exam

# Exams

- ▶ **Midterm**

Fri, Mar. 9, 2018, 4:10pm - 5:25pm, *in lecture*

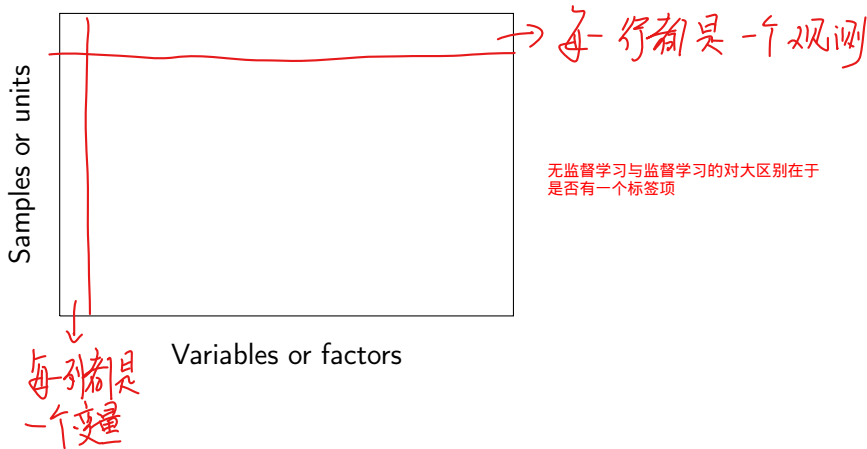
- ▶ **Final**

Fri, May 4, 2018, 1:10pm - 4:00pm, *in lecture*

- ▶ 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:





## Supervised vs. unsupervised learning

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



Variables or factors

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

## Supervised vs. unsupervised learning

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



Variables or factors

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

# Supervised vs. unsupervised learning

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

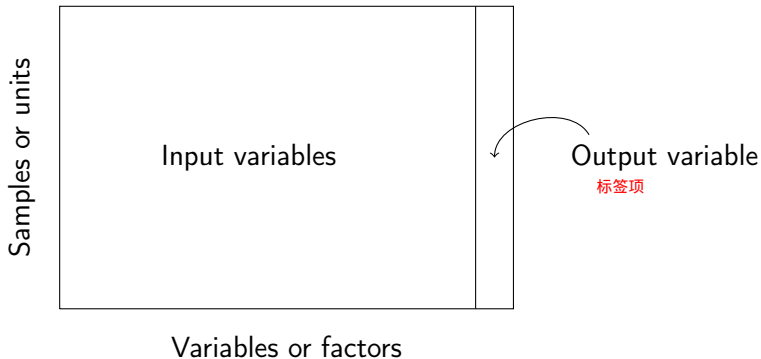
Our goal is to:

- ▶ Find meaningful relationships between the variables or units.  
Correlation analysis. 相关分析
- ▶ 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. 降维
- ▶ Find meaningful groupings of the data. Clustering. 聚类

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:

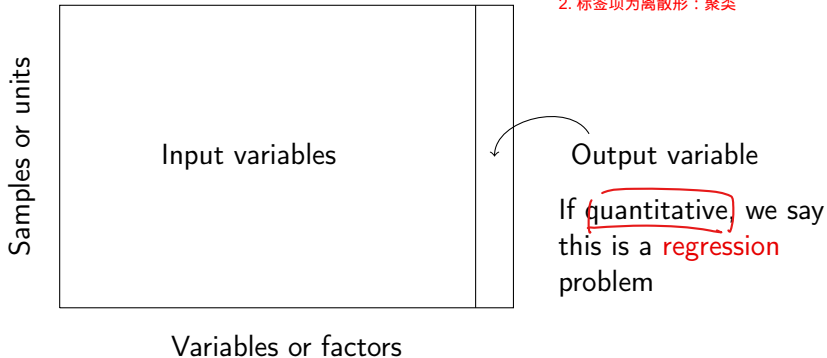


# Supervised vs. unsupervised learning

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

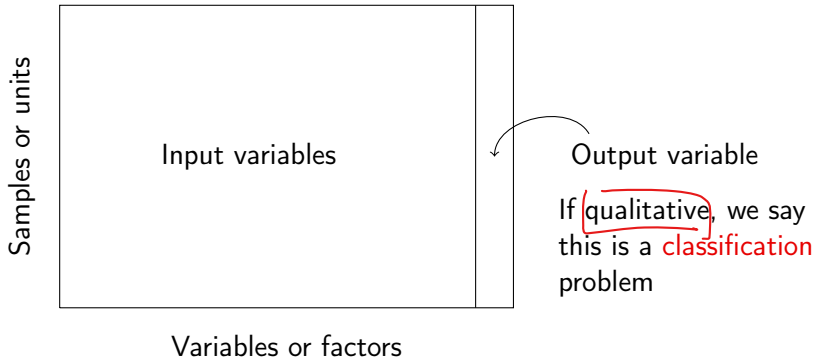
监督学习分为两个主要的类别:

1. 标签项为连续形: 回归
2. 标签项为离散形: 聚类



## 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:

★最大特点

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.

} training set: estimate the function  $f$   
} testing set: value the performance of  $f$

## 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:

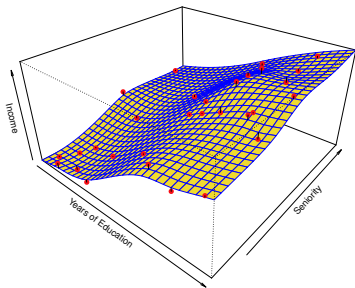
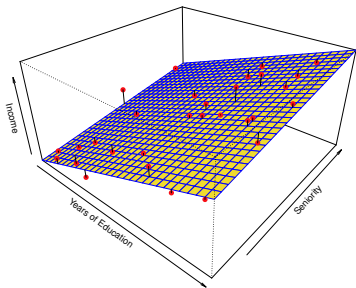
- ▶ **Parametric methods:** We assume that  $f$  takes a specific form. For example, a linear form:

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

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

↑ 易于解决 inference 问题, 但 predict 的精度低

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

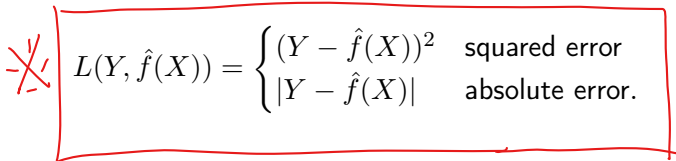
Parametric methods are often simpler to interpret.

易于不断提升 predict  
但形式复杂难以 inference

## 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:


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

## 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*.

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

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:

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

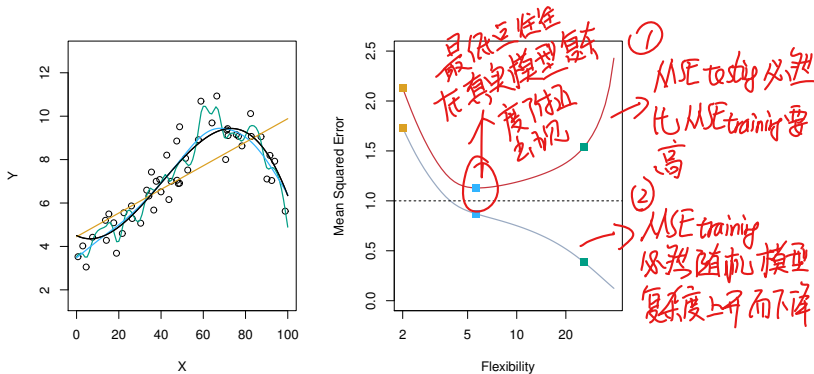
## Prediction error

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

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

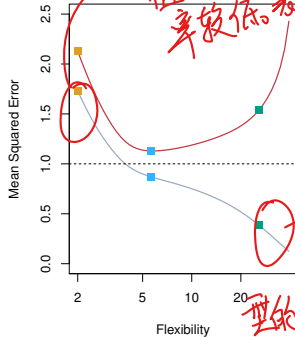
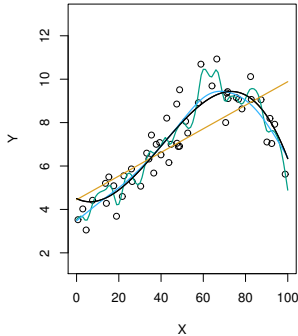
$$MSE_{\text{test}}(\hat{f}) = \frac{1}{m} \sum_{i=1}^m (y'_i - \hat{f}(x'_i))^2.$$

ISL Figure 2.9.



The circles are simulated data from the black curve.

ISL Figure 2.9.



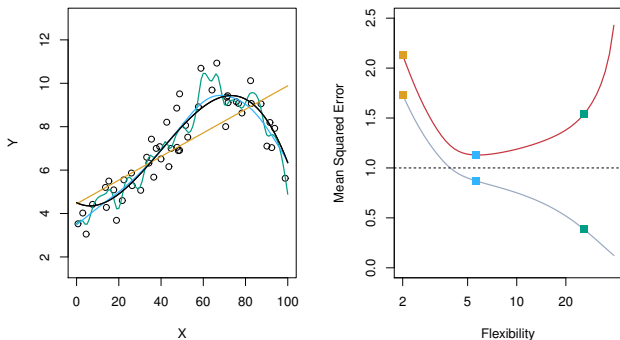
MSE training  
值较大表明模型的准确  
率较低。表明 high bias

MSE training 较  
小  $\Rightarrow$  表明对  
这一组数据模  
型的准确度较高。  
 $\Rightarrow$  易受到数据影响  
预测结果不稳定  
 $\Rightarrow$  high variance

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



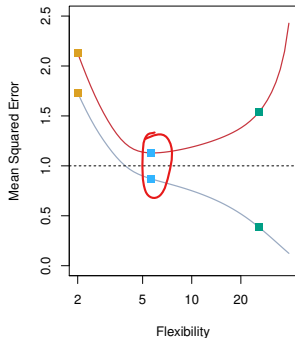
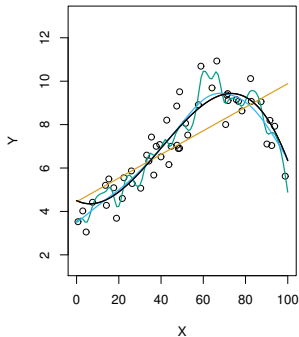
ISL Figure 2.9.



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

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

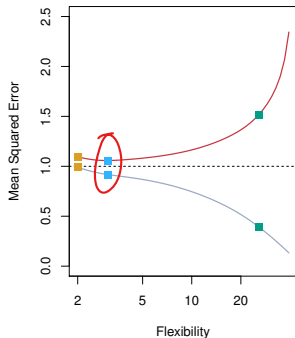
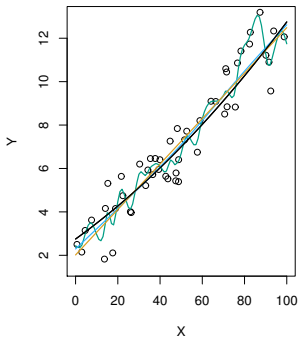
ISL Figure 2.9.



当真实模型中的  
数据复杂度  
为中等时

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

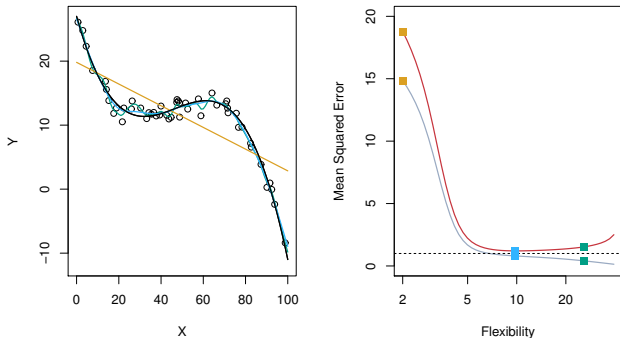
ISL Figure 2.10



当数据本身就比较为线性, 模型简单已足以有较好的精度

The function  $f$  is now almost linear.

ISL Figure 2.11



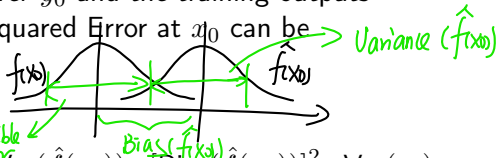
low variance of irreducible term  $\Rightarrow$  lower probability to be overfit  $\Rightarrow$  low variance of the model

When the noise  $\varepsilon$  has small variance, the third method does well.

## 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  $(x_1, y_1) \dots (x_n, y_n)$ .

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



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

$$= E[(y_0 - f(x_0)) + (f(x_0) - E(\hat{f}(x_0))) + (E(\hat{f}(x_0)) - \hat{f}(x_0))]^2$$

$$= E\left[\underbrace{(y_0 - f(x_0))}^{\text{irreducible error}} + \underbrace{(f(x_0) - E(\hat{f}(x_0)))}_{\text{Bias}(\hat{f}(x_0))} + \underbrace{(E(\hat{f}(x_0)) - \hat{f}(x_0))}_{\text{Variance}(\hat{f}(x_0))} + \text{交叉项}\right]$$

$$= E[y_0 - f(x_0)]^2$$

$$E[f(x_0) - E(\hat{f}(x_0))]^2$$

$$E[E(\hat{f}(x_0)) - \hat{f}(x_0)]^2$$

$$= \text{Var}(\varepsilon_0) \text{ [irreducible error]}$$

$$\text{Bias}(\hat{f}(x_0))^2$$

$$\text{Variance}(\hat{f}(x_0))$$

交叉项为0:

## The bias variance decomposition

$$\begin{aligned} & \hat{E}[\hat{E}(f(x_0)) - f(x_0)] \\ & \times E(y_0 - f(x_0)) \\ & = 0 \end{aligned}$$

$$\begin{aligned} & E(y_0 - f(x_0)) E(f(x_0) - \hat{E}(\hat{f}(x_0))) \\ & = E(y_0 - f(x_0)) E(f(x_0) - f(x_0)) = 0 \end{aligned}$$

$$\begin{aligned} & E(y_0 - f(x_0)) E[\hat{E}(f(x_0)) - \hat{f}(x_0)] \\ & \stackrel{||}{=} 0 \quad \stackrel{||}{=} 0 \end{aligned}$$

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

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

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

Irreducible error

## 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  $(x_1, y_1) \dots (x_n, y_n)$ .

Let  $E$  denote the expectation over  $y_0$  and the training outputs  $(y_1, \dots, y_n)$ . 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.

预测的稳定性

## 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  $(x_1, y_1) \dots (x_n, y_n)$ .

Let  $E$  denote the expectation over  $y_0$  and the training outputs  $(y_1, \dots, y_n)$ . 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 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  $f(x_0)$ .

预测的平均准确性



## Implications of bias variance decomposition

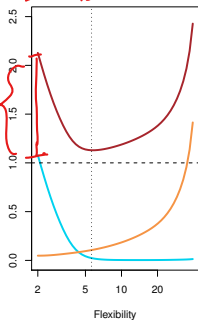
$$MSE(x_0) = E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon).$$

- ▶ 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.

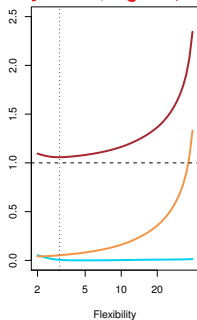
Squiggly  $f$ , high noise

数据本身规律较复杂  
且变动较大



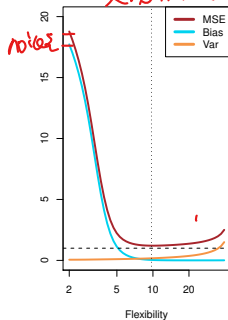
Linear  $f$ , high noise

数据本身较为简单  
且变动较大



Squiggly  $f$ , low noise

数据本身较为复杂但  
变动不大



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  $\{\text{Ford, Toyota, Mercedes-Benz, } \dots\}$ .

# 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  $\{\text{Ford, Toyota, Mercedes-Benz, } \dots\}$ .

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  $\{\text{Ford, Toyota, Mercedes-Benz, } \dots\}$ .

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  $\{\text{Ford, Toyota, Mercedes-Benz, } \dots\}$ .

We will use slightly different notation:

$P(X, Y)$  : joint distribution of  $(X, Y)$ ,  
 $P(Y | X)$  : conditional distribution of  $X$  given  $Y$ ,  
 $\hat{y}_i$  : prediction for  $x_i$ .

## 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:

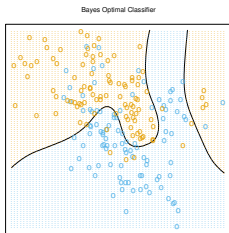
$$L_{01} = E(\mathbf{1}(y_0 \neq \hat{y}_0))$$

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

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

# 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:

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

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



proof that under 0-1 Loss, bayes classifier is optimal:

$$R(f|X) = \sum E(\mathbb{I}(y_0 \neq \hat{y}_0) | P(y|X))$$

$$= \sum_{y \neq f(x)} 1 \times p(y|x) + \sum_{y=f(x)} 0 \times p(y|x)$$

$$= \sum_{y \neq f(x)} p(y|x) = 1 - p(f(x)|X)$$

$$f_0 = \arg \max_{[k]} p(y=k|X=x_i)$$

(贝叶斯定理)

$$\Rightarrow f_0 \text{ minimize } R(f|x)$$

Thanks to Sergio Bacallado and Peter Orbanz

for sharing the slides.

$$\Rightarrow R(f) = \int R(f|x) p(x) dx$$

$$\Rightarrow f_0 \text{ minimize } R(f)$$

where  $R(f)$  is the expectation of 0-1 Loss.

给定  $x_1$ ,  
在  $y=f(x)$  的  
概率加上  $p(y)$  在  $y \neq f(x)$   
的概率为 1