High-Dimensional Statistics and Generalized Linear Models

TrungTin Nguyen

STATIFY team, Inria centre at the University Grenoble Alpes, France

Statistical Analysis and Document Mining

Complementary Course, Master of Applied Mathematics in Grenoble

Outline

Outline

Previous episode: multiple logistic regression

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in [2] \equiv \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.

Previous episode: multiple logistic regression

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in [2] \equiv \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?

Previous episode: multiple logistic regression

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in [2] \equiv \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- Multiple logistic regression (MLR) takes the form $p_2(\mathbf{X}) = 1 - p_1(\mathbf{X})$, and models the probability that Y belongs to a particular category instead Y,

$$\log\left(\frac{p_1(\mathbf{X})}{1 - p_1(\mathbf{X})}\right) = \beta_0 + \sum_{p=1}^{P} \beta_p X_p, \text{ or } p_1(\mathbf{X}) = \frac{\exp(\beta_0 + \sum_{p=1}^{P} \beta_p X_p)}{1 + \exp(\beta_0 + \sum_{p=1}^{P} \beta_p X_p)}.$$

Previous episode: multiple logistic regression

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in [2] \equiv \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- Multiple logistic regression (MLR) takes the form $p_2(\mathbf{X}) = 1 - p_1(\mathbf{X})$, and models the probability that Y belongs to a particular category instead Y,

$$\log\left(\frac{p_1(\mathbf{X})}{1-p_1(\mathbf{X})}\right) = \beta_0 + \sum_{p=1}^{P}\beta_p X_p, \text{ or } p_1(\mathbf{X}) = \frac{\exp(\beta_0 + \sum_{p=1}^{P}\beta_p X_p)}{1+\exp(\beta_0 + \sum_{p=1}^{P}\beta_p X_p)}.$$

- **Training error:** using non-linear LS or maximum likelihood estimation (MLE), we obtain $\widehat{\beta}$ and $\widehat{r}_{\mathcal{D}}(\mathbf{x}_n) = \operatorname{argmax}_{c\in\mathcal{C}} p_c(\mathbf{x}_n)$ such that $\forall n \in [N]$,

$$y_n \approx \widehat{r}_{\mathcal{D}}(\mathbf{x}_n), \text{ or equivalent, } \mathcal{L}(\widehat{r}_{\mathcal{D}}, \mathcal{D}) \equiv \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}\left[y_n \neq \widehat{r}_{\mathcal{D}}(\mathbf{x}_n)\right] \approx 0. \quad (1)$$

Previous episode: multiple logistic regression

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in [2] \equiv \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- Multiple logistic regression (MLR) takes the form $p_2(\mathbf{X}) = 1 - p_1(\mathbf{X})$, and models the probability that Y belongs to a particular category instead Y,

$$\log\left(\frac{p_1(\mathbf{X})}{1-p_1(\mathbf{X})}\right) = \beta_0 + \sum_{p=1}^{P}\beta_p X_p, \text{ or } p_1(\mathbf{X}) = \frac{\exp(\beta_0 + \sum_{p=1}^{P}\beta_p X_p)}{1+\exp(\beta_0 + \sum_{p=1}^{P}\beta_p X_p)}.$$

- **Training error:** using non-linear LS or maximum likelihood estimation (MLE), we obtain $\widehat{\beta}$ and $\widehat{r}_{\mathcal{D}}(\mathbf{x}_n) = \operatorname{argmax}_{c\in\mathcal{C}} p_c(\mathbf{x}_n)$ such that $\forall n \in [N]$,

$$y_n \approx \widehat{r}_{\mathcal{D}}(\mathbf{x}_n), \text{ or equivalent, } \mathcal{L}(\widehat{r}_{\mathcal{D}}, \mathcal{D}) \equiv \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}\left[y_n \neq \widehat{r}_{\mathcal{D}}(\mathbf{x}_n)\right] \approx 0. \quad (1)$$

- **Test (generalization) error:** for any **new sample** $(\mathbf{x}^*, y^*)$, how we guarantee

$$y^* \approx \widehat{r}_{\mathcal{D}}(\mathbf{x}^*), \text{ or equivalent, } \mathcal{L}(\widehat{r}_{\mathcal{D}}) \equiv \mathbb{E}_{\mathbf{X},Y}\left[\mathbb{1}\left(Y \neq \widehat{r}_{\mathcal{D}}(\mathbf{X})\right)\right] \approx 0? \quad (2)$$

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.
2. For each $k \in [K]$, fit a model $\widehat{r}^{(-k)}(\mathbf{x}, \mathbf{m})$, indexed by a tuning parameter (or a model) $\mathbf{m} \in \mathcal{M}$, on all samples from the training set except those in the kth fold.

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.
2. For each $k \in [K]$, fit a model $\widehat{r}^{(-k)}(\mathbf{x}, \mathbf{m})$, indexed by a tuning parameter (or a model) $\mathbf{m} \in \mathcal{M}$, on all samples from the training set except those in the kth fold.
3. Estimating test error $\mathcal{L}(\widehat{r}_{\mathcal{D}})$ via averaging the final resulting MSE estimates

$$\mathcal{L}(\widehat{r}_{\mathcal{D}}) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K}}_{\text{average over } K \text{ folds}} \underbrace{\left[\frac{1}{N_k}\sum_{n\in\mathcal{D}_k} \mathbb{1}\left(y_n \neq \widehat{r}^{(-k)}(\mathbf{x}_n)\right)\right]}_{\text{Estimate test error for each fold}} \equiv \mathrm{CV}(\widehat{r}, K, \mathbf{m}).$$

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.
2. For each $k \in [K]$, fit a model $\widehat{r}^{(-k)}(\mathbf{x}, \mathbf{m})$, indexed by a tuning parameter (or a model) $\mathbf{m} \in \mathcal{M}$, on all samples from the training set except those in the kth fold.
3. Estimating test error $\mathcal{L}(\widehat{r}_{\mathcal{D}})$ via averaging the final resulting MSE estimates

$$\mathcal{L}(\widehat{r}_{\mathcal{D}}) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K}}_{\text{average over } K \text{ folds}} \underbrace{\left[\frac{1}{N_k}\sum_{n\in\mathcal{D}_k} \mathbb{1}\left(y_n \neq \widehat{r}^{(-k)}(\mathbf{x}_n)\right)\right]}_{\text{Estimate test error for each fold}} \equiv \mathrm{CV}(\widehat{r}, K, \mathbf{m}).$$

4. Best data-driven model: $\widehat{\mathbf{m}} \equiv \mathrm{argmin}_{\mathbf{m}\in\mathcal{M}} \mathrm{CV}(\widehat{r}, K, \mathbf{m})$.

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.
2. For each $k \in [K]$, fit a model $\widehat{r}^{(-k)}(\mathbf{x}, \mathbf{m})$, indexed by a tuning parameter (or a model) $\mathbf{m} \in \mathcal{M}$, on all samples from the training set except those in the kth fold.
3. Estimating test error $\mathcal{L}(\widehat{r}_{\mathcal{D}})$ via averaging the final resulting MSE estimates

$$\mathcal{L}(\widehat{r}_{\mathcal{D}}) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K}}_{\text{average over } K \text{ folds}} \underbrace{\left[\frac{1}{N_k}\sum_{n \in \mathcal{D}_k} \mathbb{1}\left(y_n \neq \widehat{r}^{(-k)}(\mathbf{x}_n)\right)\right]}_{\text{Estimate test error for each fold}} \equiv \mathrm{CV}(\widehat{r}, K, \mathbf{m}).$$

4. Best data-driven model: $\widehat{\mathbf{m}} \equiv \operatorname{argmin}_{\mathbf{m} \in \mathcal{M}} \mathrm{CV}(\widehat{r}, K, \mathbf{m})$.
   K = ?

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.
2. For each $k \in [K]$, fit a model $\widehat{r}^{(-k)}(\mathbf{x}, \mathbf{m})$, indexed by a tuning parameter (or a model) $\mathbf{m} \in \mathcal{M}$, on all samples from the training set except those in the kth fold.
3. Estimating test error $\mathcal{L}(\widehat{r}_{\mathcal{D}})$ via averaging the final resulting MSE estimates

$$\mathcal{L}(\widehat{r}_{\mathcal{D}}) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K}}_{\text{average over } K \text{ folds}} \underbrace{\left[\frac{1}{N_k}\sum_{n\in\mathcal{D}_k} \mathbb{1}\left(y_n \neq \widehat{r}^{(-k)}(\mathbf{x}_n)\right)\right]}_{\text{Estimate test error for each fold}} \equiv \mathrm{CV}(\widehat{r}, K, \mathbf{m}).$$

4. Best data-driven model: $\widehat{\mathbf{m}} \equiv \operatorname{argmin}_{\mathbf{m}\in\mathcal{M}} \mathrm{CV}(\widehat{r}, K, \mathbf{m})$.
   $K = ?$ Setting $K = N$ yields N-fold or leave-one out cross-validation (LOOCV, **high variance**).

Previous episode: Cross-Validation for MLR

1. Split the training dataset randomly into K folds so that we have $\mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K = \mathcal{D}$, where $\mathcal{D}_k$ denotes the indices of the observations in part k. There are N_k observations in part k: if N is a multiple of K, then $N_k = N/K$.
2. For each $k \in [K]$, fit a model $\widehat{r}^{(-k)}(\mathbf{x}, \mathbf{m})$, indexed by a tuning parameter (or a model) $\mathbf{m} \in \mathcal{M}$, on all samples from the training set except those in the kth fold.
3. Estimating test error $\mathcal{L}(\widehat{r}_{\mathcal{D}})$ via averaging the final resulting MSE estimates

$$\mathcal{L}(\widehat{r}_{\mathcal{D}}) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K}}_{\text{average over } K \text{ folds}} \underbrace{\left[\frac{1}{N_k}\sum_{n \in \mathcal{D}_k} \mathbb{1}\left(y_n \neq \widehat{r}^{(-k)}(\mathbf{x}_n)\right)\right]}_{\text{Estimate test error for each fold}} \equiv \mathrm{CV}(\widehat{r}, K, \mathbf{m}).$$

4. Best data-driven model: $\widehat{\mathbf{m}} \equiv \operatorname{argmin}_{\mathbf{m} \in \mathcal{M}} \mathrm{CV}(\widehat{r}, K, \mathbf{m})$.
   $K = ?$ Setting $K = N$ yields N-fold or leave-one out cross-validation (LOOCV, **high variance**). A common better choice $K = 5$ or 10.

Outline

Previous episode: Nearest-neighbour methods

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C} \subsetneq \mathbb{N}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.

Previous episode: Nearest-neighbour methods

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C} \subsetneq \mathbb{N}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?

Previous episode: Nearest-neighbour methods

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C} \subsetneq \mathbb{N}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- K-Nearest-neighbours (kNN) approximate the probability that Y belongs to a particular category instead Y,

$$\mathbb{P}(Y = c|\mathbf{X} = \mathbf{x}) \approx \frac{1}{K} \sum_{\mathbf{x}_n \in \mathcal{N}_K(\mathbf{x})} \mathbb{1}_{y=c}(y_n).$$

Previous episode: Nearest-neighbour methods

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C} \subsetneq \mathbb{N}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- K-Nearest-neighbours (kNN) approximate the probability that Y belongs to a particular category instead Y,

$$\mathbb{P}(Y = c|\mathbf{X} = \mathbf{x}) \approx \frac{1}{K} \sum_{\mathbf{x}_n \in \mathcal{N}_K(\mathbf{x})} \mathbb{1}_{y=c}(y_n).$$

- kNN is a **nonparametric classifier**: it makes **no assumptions about the data**. $\rightarrow$ It can be **very flexible** but may **not be optimal** when we **know something about the data.**

Previous episode: Nearest-neighbour methods

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C} \subsetneq \mathbb{N}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- K-Nearest-neighbours (kNN) approximate the probability that Y belongs to a particular category instead Y,

$$\mathbb{P}(Y = c|\mathbf{X} = \mathbf{x}) \approx \frac{1}{K} \sum_{\mathbf{x}_n \in \mathcal{N}_K(\mathbf{x})} \mathbb{1}_{y=c}(y_n).$$

- kNN is a **nonparametric classifier**: it makes **no assumptions about the data**. $\rightarrow$ It can be **very flexible** but may **not be optimal** when we **know something about the data.**
- The hyper-parameter K usually chosen via cross-validation.

It works well in low dimensions, but suffers from the curse of dimensionality. To be verified soon!

Outline

1. Previous Episode: Discriminative Approaches for Classification Problems
   - Multiple logistic regression
   - Nearest-neighbour methods
   - High-dimensional data classification
2. Curse of Dimensionality
   - A synthetically generated Oil Flow data set
   - KNN learning algorithm
   - Curse of dimensionality for kNN
   - Lost in the immensity of high-dimensional spaces
   - The impact of high dimensionality on statistics is multiple!
3. Generalized Linear Models
   - Bikeshare data set: motivation example
   - Bikeshare data: linear regression
   - Bikeshare data: Poisson regression
   - Bikeshare data: linear regression and Poisson regression
   - Generalized linear models

High-dimensional data classification

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.

High-dimensional data classification

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- In genomics and other areas of computational biology, the number of features P is much larger than the number of observations N, often written as $P \gg N$, where $\mathbf{x}_n \in \mathbb{R}^P$.

High-dimensional data classification

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- In genomics and other areas of computational biology, the number of features P is much larger than the number of observations N, often written as $P \gg N$, where $\mathbf{x}_n \in \mathbb{R}^P$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?

High-dimensional data classification

- We are given a **training dataset** $\mathcal{D} \equiv \{(\mathbf{x}_n, y_n)\}_{n\in[N]}$, $y_{[N]} \in \mathcal{C}$, i.i.d. sampled from **the true (but unknown) joint PDF** of $(\mathbf{X}, Y)$.
- In genomics and other areas of computational biology, the number of features P is much larger than the number of observations N, often written as $P \gg N$, where $\mathbf{x}_n \in \mathbb{R}^P$.
- How to model relationship between $p_c(\mathbf{X}) = \mathbb{P}(Y = c|\mathbf{X})$, $c \in \mathcal{C}$, and $\mathbf{X}$?
- Which is suitable for high-dimensional data: Discriminative approaches (CM5) or Generative approaches (CM6).
  1. K-nearest neighbors (K-NN) (CM5),
  2. Logistic Regression (CM5),
  3. Linear Discriminant Analysis (CM5),
  4. Naive Bayes classifier (CM6).

Outline

Outline

A synthetically generated Oil Flow data set[1]

- Representing $N = 1000$ **data points** (for each training, validation, and test sets) taken from a pipeline containing a mixture of oil, water, and gas.

[1] Bishop, C. M. and G. D. James (1993). *Analysis of multiphase flows using dual-energy gamma densitometry and neural networks.* Nuclear Instruments and Methods in Physics Research.

A synthetically generated Oil Flow data set[1]

- Representing $N = 1000$ **data points** (for each training, validation, and test sets) taken from a pipeline containing a mixture of oil, water, and gas.
- Comprising a $P = 12$-**dimensional input vector** consisting of measurements taken with gamma ray densitometers that measure the attenuation of gamma rays passing along narrow beams through the pipe.
- Each data point is labelled according to which of the three geometrical classes it belongs to: $Y \in \{$Stratified, Annular, Homogeneous$\}$, see Figure 1 for more details.

[1] Bishop, C. M. and G. D. James (1993). *Analysis of multiphase flows using dual-energy gamma densitometry and neural networks.* Nuclear Instruments and Methods in Physics Research.

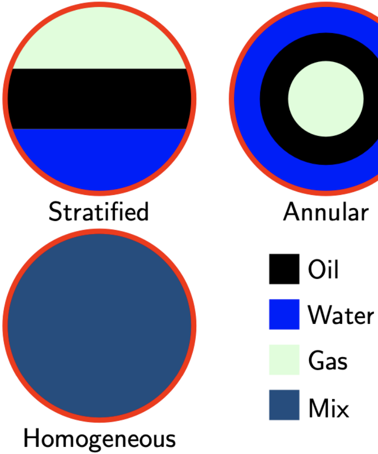


Figure 1: The three geometrical configurations of the oil, water, and gas phases used to generate the oil-flow data set. For each configuration, the proportions of the three phases can vary [Bishop, 2006, Figure A.2].

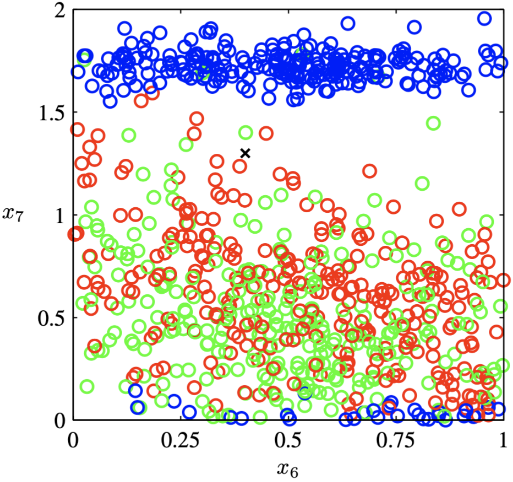
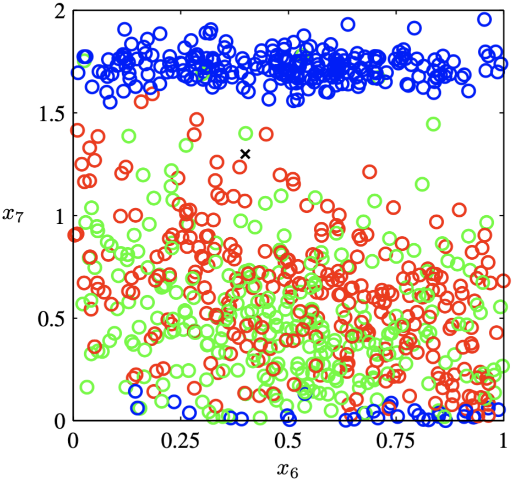
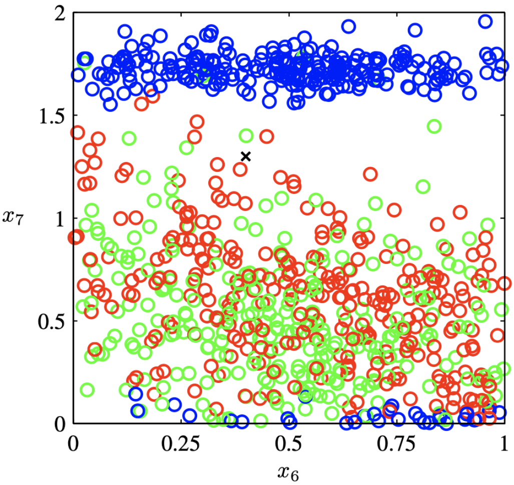


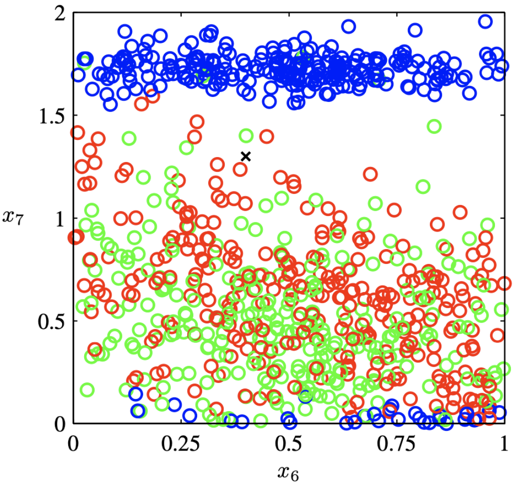
Figure 2: Scatter plot of the oil flow data for input variables x_6 and x_7, in which red denotes the 'homogenous' class, green denotes the 'annular' class, and blue denotes the 'laminar' class. **Our goal is to classify the new test point denoted by '×'.** [Bishop, 2006, Figure 1.19].



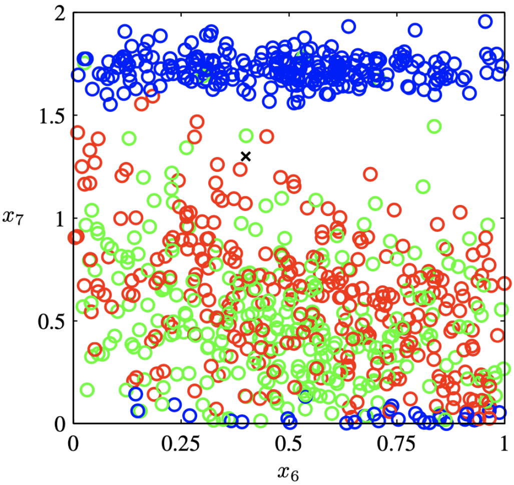
❓ **Your intuitive solution ...** New test point denoted by '×' should be x, x or x?



$\boxed{\textbf{Intuitive solution}}$ the cross $\mathbf{x}$ is surrounded by numerous red points, and so we might suppose that it belongs to the red class.

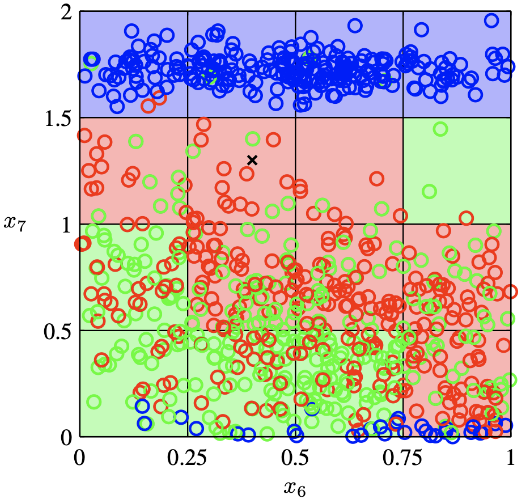


$\boxed{\textbf{Intuitive solution}}$ the cross **x** is surrounded by numerous red points, and so we might suppose that it belongs to the red class. There are also plenty of green points nearby, so we might think that it could instead belong to the green class.

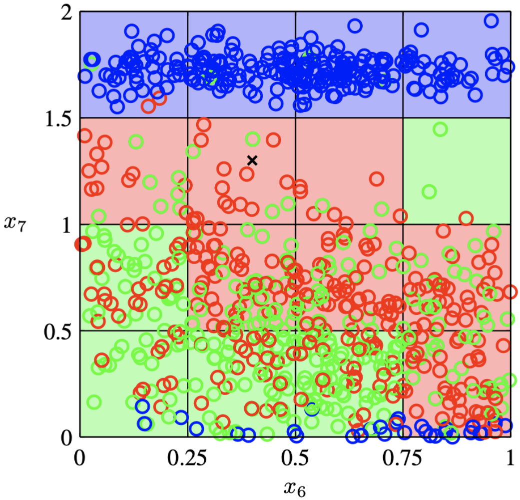


$\boxed{\textbf{Intuitive solution}}$ the cross $\mathbf{x}$ is surrounded by numerous red points, and so we might suppose that it belongs to the red class. There are also plenty of green points nearby, so we might think that it could instead belong to the green class. It seems unlikely that it belongs to the blue class.

Outline

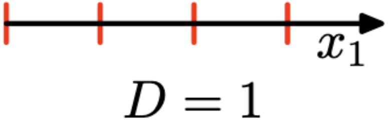


KNN learning algorithm: **divide the input space into regular cells** and any new test point is assigned to the class that has a **majority number of representatives in the same cell as the test point**. [Bishop, 2006, Figure 1.20].

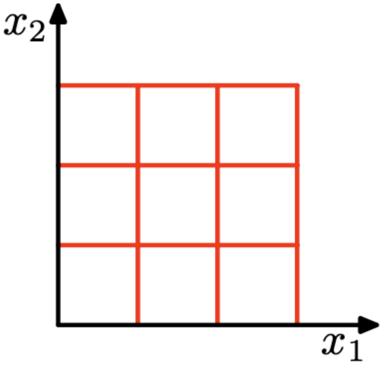


$\boxed{\textbf{KNN learning algorithm:}}$ **divide the input space into regular cells** and any new test point is assigned to the class that has a **majority number of representatives in the same cell as the test point**. [Bishop, 2006, Figure 1.20]. ❓A few potentially serious shortcomings . . .

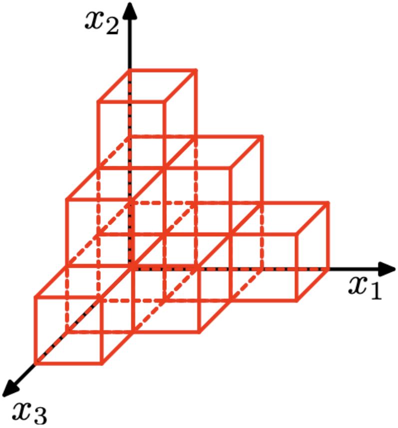
Outline



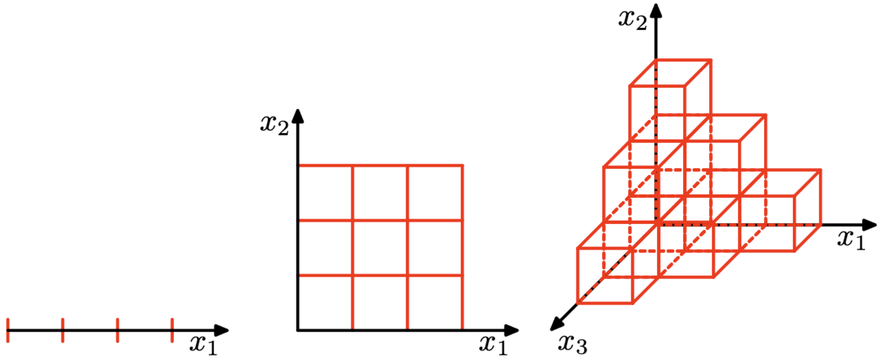
$\boxed{\textbf{Illustration of the curse of dimensionality:}}$ number of cells $= 3^1 = 3$ [Bishop, 2006, Figure 1.21].



$\boxed{\textbf{Illustration of the curse of dimensionality:}}$ when $P = 2$, number of cells $= 3^2 = 9$ [Bishop, 2006, Figure 1.21].



$\boxed{\textbf{Illustration of the curse of dimensionality:}}$ when $P = 3$, number of cells $= 3^3 = 27$ [Bishop, 2006, Figure 1.21].



$\boxed{\textbf{Illustration of the curse of dimensionality:}}$ **exponentially large number of cells** 3^P, ***e.g.***, $3^{12} = 531441$ is that we would need an **exponentially large quantity of training data** in order to ensure that the **cells are not empty** [Bishop, 2006, Figure 1.21].

Outline

Lost in the immensity of high-dimensional spaces

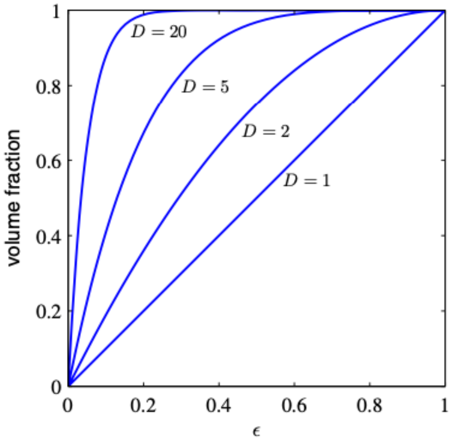
☛ Our geometrical intuitions, formed through a life spent in a space of three dimensions, can fail badly when we consider spaces of higher dimensionality [Bishop, 2006].

Our geometrical intuitions, formed through a life spent in a space of three dimensions, can fail badly when we consider spaces of higher dimensionality [Bishop, 2006].

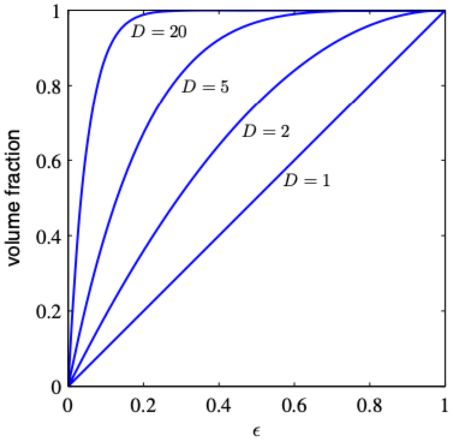
- Fraction of the volume of the sphere that lies between radius $r = 1 - \epsilon$ and $r = 1$ is given by

$$\frac{V_P(1) - V_P(1-\epsilon)}{V_P(1)} = 1 - (1-\epsilon)^P \longrightarrow 1 \text{ for large } P \text{ and small } \epsilon!$$

Here, $V_P(1)$ is a volume of a sphere of radius r in $P(\equiv D)$ dimensions.



Plot of the fraction of the volume of a sphere that lies between radius $r = 1 - \epsilon$ and $r = 1$ for various values of the dimensionality $P (\equiv D)$ [Bishop, 2006, Figure 1.22]. ANY comments ?



Plot of the fraction of the volume of a sphere that lies between radius $r = 1 - \epsilon$ and $r = 1$ for various values of the dimensionality $P (\equiv D)$ [Bishop, 2006, Figure 1.22]. ANY comments ?

In spaces of high dimensionality, most of the volume of a sphere is concentrated in a thin shell near the surface!

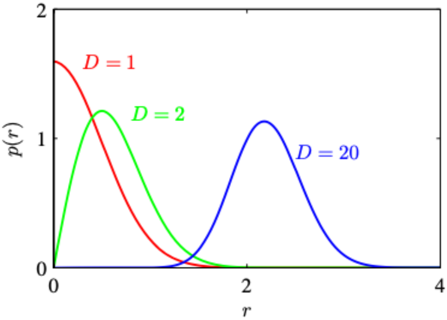
Lost in the immensity of high-dimensional spaces

☛ Our geometrical intuitions, formed through a life spent in a space of three dimensions, can fail badly when we consider spaces of higher dimensionality [Bishop, 2006].

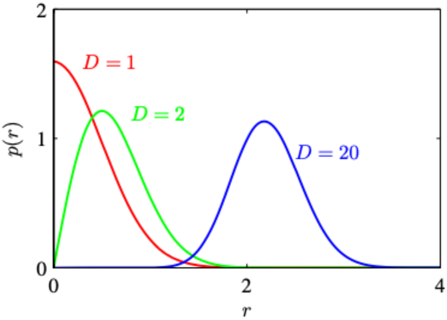
Lost in the immensity of high-dimensional spaces

Our geometrical intuitions, formed through a life spent in a space of three dimensions, can fail badly when we consider spaces of higher dimensionality [Bishop, 2006].

- Behaviour of a Gaussian distribution in a high-dimensional space.
- If we transform from Cartesian to polar coordinates, and then integrate out the directional variables, we obtain an expression for the density $p(r)$ as a function of radius r from the origin.
- Thus $p(r)\delta r$ is the probability mass inside a thin shell of thickness δr located at radius r.



Plot of the probability density with respect to radius r of a Gaussian distribution for various values of the dimensionality $P(\equiv D)$. [Bishop, 2006, Figure 1.23]. ANY comments ?



Plot of the probability density with respect to radius r of a Gaussian distribution for various values of the dimensionality $P(\equiv D)$. [Bishop, 2006, Figure 1.23]. ANY comments ❓

For large $P(\equiv D)$ most of the probability mass of a Gaussian is concentrated in a thin shell!

Lost in the immensity of high-dimensional spaces

☛ High-dimensional spaces are vast and data points are isolated in their immensity [Giraud, 2021].

- So far, we have considered the situation where we want to explain a response variable $Y \in \mathbb{R}$ by P variables $X_1, ..., X_P \in [0, 1]$. Assume that each variable X_p follows a uniform distribution on $[0, 1]$.

Lost in the immensity of high-dimensional spaces

High-dimensional spaces are vast and data points are isolated in their immensity [Giraud, 2021].

- So far, we have considered the situation where we want to explain a response variable $Y \in \mathbb{R}$ by P variables $X_1, \dots, X_P \in [0, 1]$. Assume that each variable X_p follows a uniform distribution on $[0, 1]$.
- If these variables are independent, then the variable $X = (X_1, \dots, X_p) \in [0, 1]^P$ follows a uniform distribution on the hypercube $[0, 1]^P$.

Lost in the immensity of high-dimensional spaces

☛ High-dimensional spaces are vast and data points are isolated in their immensity [Giraud, 2021].

- So far, we have considered the situation where we want to explain a response variable $Y \in \mathbb{R}$ by P variables $X_1, ..., X_P \in [0, 1]$. Assume that each variable X_p follows a uniform distribution on $[0, 1]$.
- If these variables are independent, then the variable $X = (X_1, ..., X_p) \in [0, 1]^P$ follows a uniform distribution on the hypercube $[0, 1]^P$.
- Consider classical regression, given $\epsilon_{[N]}$ independent and centered,

$$Y_n = f(\mathbf{X}_n) + \epsilon_n, \quad n \in [N] \text{ with } f : [0, 1]^P \to \mathbb{R}. \tag{3}$$

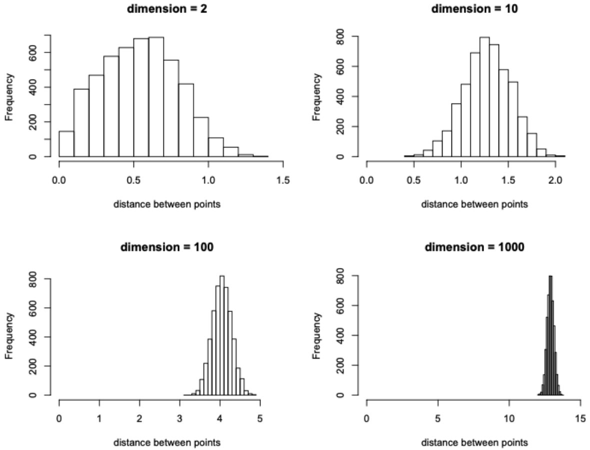
Lost in the immensity of high-dimensional spaces

High-dimensional spaces are vast and data points are isolated in their immensity [Giraud, 2021].

- So far, we have considered the situation where we want to explain a response variable $Y \in \mathbb{R}$ by P variables $X_1, ..., X_P \in [0, 1]$. Assume that each variable X_p follows a uniform distribution on $[0, 1]$.
- If these variables are independent, then the variable $X = (X_1, ..., X_p) \in [0, 1]^P$ follows a uniform distribution on the hypercube $[0, 1]^P$.
- Consider classical regression, given $\epsilon_{[N]}$ independent and centered,

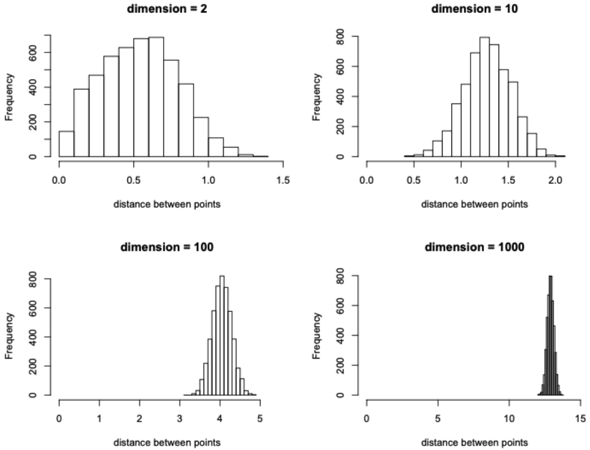
$$Y_n = f(\mathbf{X}_n) + \epsilon_n, \quad n \in [N] \text{ with } f : [0, 1]^P \to \mathbb{R}. \tag{3}$$

Assuming that the function f is smooth, it is natural to estimate $f(x)$ by some average of the Y_n associated to the $\mathbf{X}_n$ in the vicinity of x. The most simple version of this idea is the kNN estimator, where $f(x)$ is estimated by the mean of the Y_n associated to the k points $\mathbf{X}_n$, which are the nearest from x!



Histograms of the pairwise-distances between $N = 100$ points sampled uniformly in the hypercube $[0, 1]^P$, for $P = 2, 10, 100,$ and 1000 [Giraud, 2021, Figure 1.3].

ANY comments ?



Histograms of the pairwise-distances between $N = 100$ points sampled uniformly in the hypercube $[0,1]^P$, for $P = 2, 10, 100$, and 1000 [Giraud, 2021, Figure 1.3].
ANY comments ?
For large P: (1) the minimal distance between two points increases, (2) all the points are at a similar distance from the others, so the notion of "nearest points" vanishes.

Outline

1. Previous Episode: Discriminative Approaches for Classification Problems
   - Multiple logistic regression
   - Nearest-neighbour methods
   - High-dimensional data classification
2. Curse of Dimensionality
   - A synthetically generated Oil Flow data set
   - KNN learning algorithm
   - Curse of dimensionality for kNN
   - Lost in the immensity of high-dimensional spaces
   - The impact of high dimensionality on statistics is multiple!
3. Generalized Linear Models
   - Bikeshare data set: motivation example
   - Bikeshare data: linear regression
   - Bikeshare data: Poisson regression
   - Bikeshare data: linear regression and Poisson regression
   - Generalized linear models

1. High-dimensional spaces are vast and data points are isolated in their immensity.

1. High-dimensional spaces are vast and data points are isolated in their immensity.
2. The accumulation of small fluctuations in many different directions can produce a large global fluctuation.

1. High-dimensional spaces are vast and data points are isolated in their immensity.
2. The accumulation of small fluctuations in many different directions can produce a large global fluctuation.
3. An event that is an accumulation of rare events may not be rare.

1. High-dimensional spaces are vast and data points are isolated in their immensity.
2. The accumulation of small fluctuations in many different directions can produce a large global fluctuation.
3. An event that is an accumulation of rare events may not be rare.
4. Numerical computations and optimizations in high-dimensional spaces can be overly intensive.

Outline

1. Previous Episode: Discriminative Approaches for Classification Problems
   - Multiple logistic regression
   - Nearest-neighbour methods
   - High-dimensional data classification
2. Curse of Dimensionality
   - A synthetically generated Oil Flow data set
   - KNN learning algorithm
   - Curse of dimensionality for kNN
   - Lost in the immensity of high-dimensional spaces
   - The impact of high dimensionality on statistics is multiple!
3. Generalized Linear Models
   - Bikeshare data set: motivation example
   - Bikeshare data: linear regression
   - Bikeshare data: Poisson regression
   - Bikeshare data: linear regression and Poisson regression
   - Generalized linear models

Outline

Bikeshare data set: motivation example

☛ We consider the Bikeshare data set. The response is bikers, the **number of hourly users of a bike sharing** program in Washington, DC. This response value is **neither qualitative nor quantitative**: instead, it **takes on non-negative integer values, or counts.**

Bikeshare data set: motivation example

We consider the Bikeshare data set. The response is bikers, the **number of hourly users of a bike sharing** program in Washington, DC. This response value is **neither qualitative nor quantitative**: instead, it **takes on non-negative integer values, or counts.**

Predicting bikers using the covariates mnth (month of the year), hr (hour of the day, from 0 to 23), workingday (an indicator variable that equals 1 if it is neither a weekend nor a holiday), temp (the normalized temperature, in Celsius), and weathersit (a qualitative variable that takes on one of four possible values: clear; misty or cloudy; light rain or light snow; or heavy rain or heavy snow.)

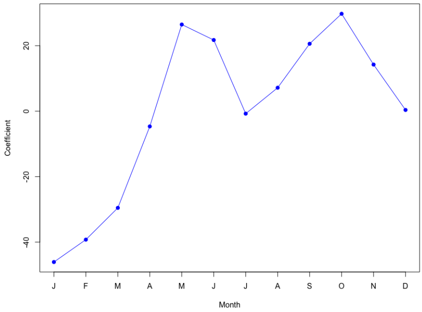
```
> head(Bikeshare)
  season mnth day hr holiday weekday workingday   weathersit temp  atemp  hum windspeed casual registered bikers
1      1  Jan   1  0       0       6          0        clear 0.24 0.2879 0.81    0.0000      3         13     16
2      1  Jan   1  1       0       6          0        clear 0.22 0.2727 0.80    0.0000      8         32     40
3      1  Jan   1  2       0       6          0        clear 0.22 0.2727 0.80    0.0000      5         27     32
4      1  Jan   1  3       0       6          0        clear 0.24 0.2879 0.75    0.0000      3         10     13
5      1  Jan   1  4       0       6          0        clear 0.24 0.2879 0.75    0.0000      0          1      1
6      1  Jan   1  5       0       6          0 cloudy/misty 0.24 0.2576 0.75    0.0896      0          1      1
```

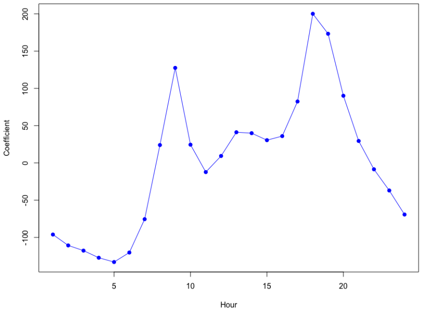
Outline

```
> mod.lm2 <- lm(
+     bikers ~ mnth + hr + workingday + temp + weathersit,
+     data = Bikeshare
+   )
> summary(mod.lm2)

Call:
lm(formula = bikers ~ mnth + hr + workingday + temp + weathersit,
    data = Bikeshare)

Residuals:
    Min      1Q  Median      3Q     Max
-299.00  -45.70   -6.23   41.08  425.29

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  73.5974     5.1322  14.340  < 2e-16 ***
mnth1                       -46.0871     4.0855 -11.281  < 2e-16 ***
mnth2                       -39.2419     3.5391 -11.088  < 2e-16 ***
mnth3                       -29.5357     3.1552  -9.361  < 2e-16 ***
mnth4                        -4.6622     2.7406  -1.701  0.08895 .
mnth5                        26.4700     2.8508   9.285  < 2e-16 ***
mnth6                        21.7317     3.4651   6.272 3.75e-10 ***
mnth7                        -0.7626     3.9084  -0.195  0.84530
mnth8                         7.1560     3.5347   2.024  0.04295 *
mnth9                        20.5912     3.0456   6.761 1.46e-11 ***
mnth10                       29.7472     2.6995  11.019  < 2e-16 ***
mnth11                       14.2229     2.8604   4.972 6.74e-07 ***
```

```
mnth11                      14.2229      2.8604   4.972 6.74e-07 ***
hr1                        -96.1420      3.9554 -24.307  < 2e-16 ***
hr2                       -110.7213      3.9662 -27.916  < 2e-16 ***
hr3                       -117.7212      4.0165 -29.310  < 2e-16 ***
hr4                       -127.2828      4.0808 -31.191  < 2e-16 ***
hr5                       -133.0495      4.1168 -32.319  < 2e-16 ***
hr6                       -120.2775      4.0370 -29.794  < 2e-16 ***
hr7                        -75.5424      3.9916 -18.925  < 2e-16 ***
hr8                         23.9511      3.9686   6.035 1.65e-09 ***
hr9                        127.5199      3.9500  32.284  < 2e-16 ***
hr10                        24.4399      3.9360   6.209 5.57e-10 ***
hr11                       -12.3407      3.9361  -3.135  0.00172 **
hr12                         9.2814      3.9447   2.353  0.01865 *
hr13                        41.1417      3.9571  10.397  < 2e-16 ***
hr14                        39.8939      3.9750  10.036  < 2e-16 ***
hr15                        30.4940      3.9910   7.641 2.39e-14 ***
hr16                        35.9445      3.9949   8.998  < 2e-16 ***
hr17                        82.3786      3.9883  20.655  < 2e-16 ***
hr18                       200.1249      3.9638  50.488  < 2e-16 ***
hr19                       173.2989      3.9561  43.806  < 2e-16 ***
hr20                        90.1138      3.9400  22.872  < 2e-16 ***
hr21                        29.4071      3.9362   7.471 8.74e-14 ***
hr22                        -8.5883      3.9332  -2.184  0.02902 *
hr23                       -37.0194      3.9344  -9.409  < 2e-16 ***
workingday                   1.2696      1.7845   0.711  0.47681
temp                       157.2094     10.2612  15.321  < 2e-16 ***
weathersitcloudy/misty     -12.8903      1.9643  -6.562 5.60e-11 ***
weathersitlight rain/snow  -66.4944      2.9652 -22.425  < 2e-16 ***
```

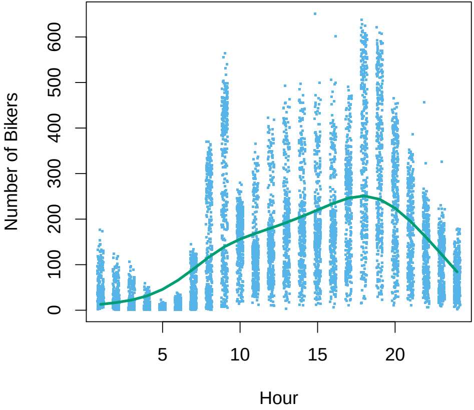
```
hr20                         90.1138     3.9400  22.872  < 2e-16 ***
hr21                         29.4071     3.9362   7.471 8.74e-14 ***
hr22                         -8.5883     3.9332  -2.184  0.02902 *
hr23                        -37.0194     3.9344  -9.409  < 2e-16 ***
workingday                    1.2696     1.7845   0.711  0.47681
temp                        157.2094    10.2612  15.321  < 2e-16 ***
weathersitcloudy/misty      -12.8903     1.9643  -6.562 5.60e-11 ***
weathersitlight rain/snow   -66.4944     2.9652 -22.425  < 2e-16 ***
weathersitheavy rain/snow -109.7446    76.6674  -1.431  0.15234
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 76.5 on 8605 degrees of freedom
Multiple R-squared:  0.6745,    Adjusted R-squared:  0.6731
F-statistic: 457.3 on 39 and 8605 DF,  p-value: < 2.2e-16
```



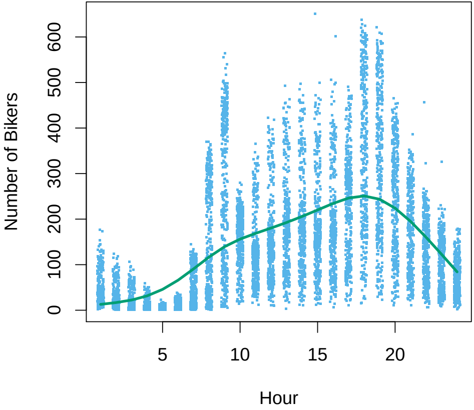


A least squares linear regression model was fit to predict bikers in the Bikeshare data set. Left: The coefficients associated with the month of the year. Bike usage is highest in the spring and fall, and lowest in the winter. Right: The coefficients associated with the hour of the day. Bike usage is highest during peak commute times, and lowest overnight [James et al., 2021, Figure 4.13]

❓ At first glance, fitting a linear regression model to the Bikeshare **seems to provide reasonable and intuitive results.**



For the most part, as the **mean number of bikers increases**, so does **the variance in the number of bikers** [James et al., 2021, Figure 4.14].



For the most part, as the **mean number of bikers increases**, so does **the variance in the number of bikers** [James et al., 2021, Figure 4.14].

❓ At first glance, fitting a linear regression model to the Bikeshare **seems to provide reasonable and intuitive results.**

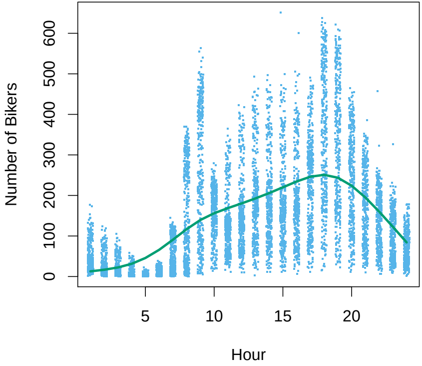
Some issues

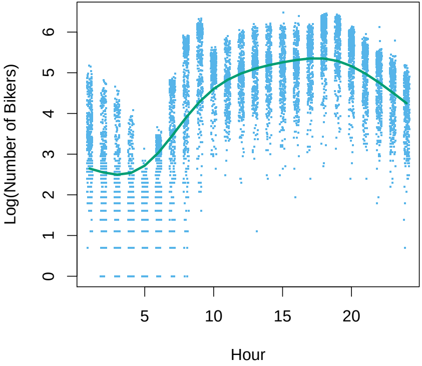
1. **9.6% of the fitted values** in the Bikeshare data set are **negative**: that is, the linear regression model predicts a negative number of users during 9.6% of the hours in the data set.

Some issues

1. **9.6% of the fitted values** in the Bikeshare data set are **negative**: that is, the linear regression model predicts a negative number of users during 9.6% of the hours in the data set.
2. Since ϵ is a continuous-valued error term, response Y is necessarily continuous-valued (quantitative) but the response bikers is integer-valued.

3. The mean-variance relationship is a major violation of the assumptions of a linear model, which state that $Y = \beta_0 + \sum_{p=1}^{P} X_p \beta_p + \epsilon$, where ϵ is a mean-zero error term with variance σ^2 that is constant, and not a function of the covariates. For the most part, as the mean number of bikers increases, so does the variance in the number of bikers!





Some solutions

1. Transforming the response Y avoids the possibility of negative predictions, and it overcomes much of the heteroscedasticity in the untransformed data $\log(Y) = \beta_0 + \sum_{p=1}^{P} X_p \beta_p + \epsilon$.

Some solutions

1. Transforming the response Y avoids the possibility of negative predictions, and it overcomes much of the heteroscedasticity in the untransformed data $\log(Y) = \beta_0 + \sum_{p=1}^{P} X_p\beta_p + \epsilon$.
   BUT Making predictions and inference in terms of the log of the response, rather than the response $\rightarrow$challenges in interpretation!

Some solutions

1. Transforming the response Y avoids the possibility of negative predictions, and it overcomes much of the heteroscedasticity in the untransformed data $\log(Y) = \beta_0 + \sum_{p=1}^{P} X_p \beta_p + \epsilon$.
   BUT Making predictions and inference in terms of the log of the response, rather than the response $\rightarrow$challenges in interpretation!

- Poisson regression model provides a much more natural and elegant approach for this task! (See more in CC6).

Some solutions

1. Transforming the response Y avoids the possibility of negative predictions, and it overcomes much of the heteroscedasticity in the untransformed data $\log(Y) = \beta_0 + \sum_{p=1}^{P} X_p \beta_p + \epsilon$.
   BUT Making predictions and inference in terms of the log of the response, rather than the response $\rightarrow$challenges in interpretation!

- Poisson regression model provides a much more natural and elegant approach for this task! (See more in CC6).
- Transforming the mean of response $\mathbb{E}[Y]$ as in the logistic regression!

Outline

1. Previous Episode: Discriminative Approaches for Classification Problems
   - Multiple logistic regression
   - Nearest-neighbour methods
   - High-dimensional data classification
2. Curse of Dimensionality
   - A synthetically generated Oil Flow data set
   - KNN learning algorithm
   - Curse of dimensionality for kNN
   - Lost in the immensity of high-dimensional spaces
   - The impact of high dimensionality on statistics is multiple!
3. Generalized Linear Models
   - Bikeshare data set: motivation example
   - Bikeshare data: linear regression
   - Bikeshare data: Poisson regression
   - Bikeshare data: linear regression and Poisson regression
   - Generalized linear models

Outline

Outline

References

Bishop, C. M. (2006).
Pattern Recognition and Machine Learning.
Information Science and Statistics. New York: Springer, 1 edition.

(Cited on pages 31, 32, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 50, 51, 52, and 53.)

Giraud, C. (2021).
Introduction to High-Dimensional Statistics.
Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

(Cited on pages 54, 55, 56, 57, 58, and 59.)

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021).
An Introduction to Statistical Learning: with Applications in R, volume 2 of *Springer Texts in Statistics*.
Springer.

(Cited on pages 73, 74, and 75.)