제8강
데이터 시각화

Section 01

데이터 시각화 기법

Section 02

ggplot 패키지

Section 03

차원 축소

3. 차원 축소

1. 차원 축소의 개념

- 산점도는 2차원 평면상에 두 변수의 값으로 좌표로 정하여 위치를 나타내는 방법으로 데이터의 분포를 관찰할 수 있는 시각화 도구
- **변수가 4개인 4차원 데이터에 대한 산점도는 어떻게 그릴 수 있을까? → 4차원을 2차원으로 축소하여 그림**
- **차원 축소(dimension reduction)란 고차원 데이터를 2,3 차원 데이터로 축소하는 기법을 말하는데, 2,3 차원으로 축소된 데이터로 산점도를 작성하여 데이터 분포를 확인하면 고차원상의 데이터 분포를 추정 가능**
- 어떻게 차원을 축소 하는가? → 3차원상의 물체에 빛을 비추면 2차원 평면에 물체의 그림자가 생기는 것과 비슷한 방법(3차원이 2차원으로 축소됨)
- 데이터의 차원을 축소하면 원래 가지고 있던 정보의 손실이 일어남

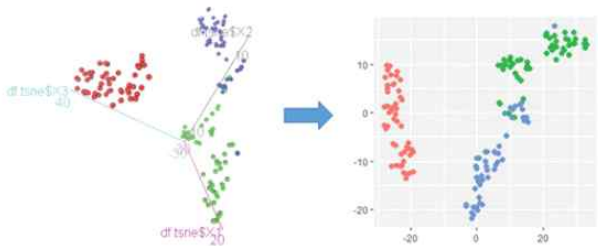


그림 8-7 3차원상의 데이터 분포를 2차원상의 분포로 변환하는 사례

3. 차원 축소

2. R을 이용한 차원 축소

2.1 4차원 데이터를 2차원 산점도로 작성하기

코드 8-14

```
library(Rtsne)
library(ggplot2)
ds <- iris[,-5]                    # 품종 정보 제외

## 중복 데이터 제거
dup = which(duplicated(ds))
dup                                # 143번째 행 중복
ds <- ds[-dup,]
ds.y <- iris$Species[-dup] # 중복을 제외한 품종 정보

## t-SNE 실행
tsne <- Rtsne(ds,dims=2, perplexity=10)

## 축소결과 시각화
df.tsne <- data.frame(tsne$Y)
head(df.tsne)
ggplot(df.tsne, aes(x=X1, y=X2, color=ds.y)) +
 geom_point(size=2)
```

3. 차원 축소

```
> library(Rtsne)
> library(ggplot2)
>
> ds <- iris[,-5]                          # 품종 정보 제외
> ## 중복 데이터 제거
> dup = which(duplicated(ds))
> dup                                      # 143번째 행 중복
[1] 143

> ds <- ds[-dup,]
> ds.y <- iris$Species[-dup]               # 중복을 제외한 품종 정보
> ## t-SNE 실행
> tsne <- Rtsne(ds,dims=2, perplexity=10)
```

t-sne를 이용하려면 중복된 데이터가 존재하면 안됨

이것을 검사하는 명령문이 which(duplicated(ds))인데, 만일 중복이 있으면 중복된 행의 번호를 dup에 보관

dup의 값을 보면 143번째 행이 중복되었다고 나오는데 실제로 143번째 행은 102번째 행과 동일

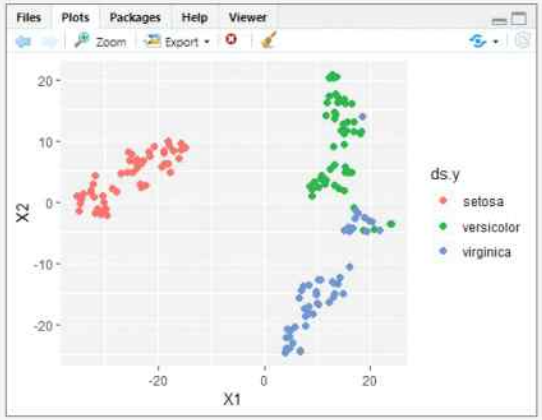
3. 차원 축소

- `ds`
  차원 축소 대상 데이터셋이다.
- `dims=2`
  ds를 몇 차원으로 축소할지 지정하는데, 2 또는 3이 일반적이다.
- `perplexity=10`
  차원 축소 과정에서 데이터를 샘플링하는데, 샘플의 개수를 몇 개로 할지 지정한다. (대상 데이터의 행의 수)/3 보다 작게 지정한다.

```
> ##  축소결과 시각화
> df.tsne <- data.frame(tsne$Y)
> head(df.tsne)
         X1        X2
1 -21.21195  6.847538
2 -14.96598  1.196396
3 -17.08658 -1.832247
4 -16.29159 -1.130622
5 -20.17702  7.507484
6 -26.10980 12.302749
```

3. 차원 축소

```
> ggplot(df.tsne, aes(x=X1, y=X2, color=ds.y)) +
+     geom_point(size=2)
>
```



3. 차원 축소

2.2 4차원 데이터를 3차원 산점도로 작성하기

코드 8-15

```
install.packages(c("rgl", "car"))
library("car")
library("rgl")
library("mgcv")

tsne <- Rtsne(ds,dims=3, perplexity=10)
df.tsne <- data.frame(tsne$Y)
head(df.tsne)

# 회귀면이 포함된 3차원 산점도
scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3)

# 회귀면이 없는 3차원 산점도
points <- as.integer(ds.y)
color <- c('red','green','blue')
scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3,
          point.col = color[points],   # 점의 색을 품종별로 다르게
          surface=FALSE)               # 회귀면을 표시하지 않음
```

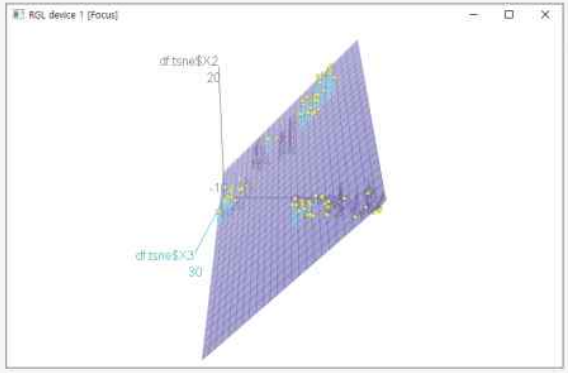
3. 차원 축소

```
> install.packages(c("rgl", "car"))
> library("car")
> library("rgl")
> library("mgcv")

tsne <- Rtsne(ds,dims=3, perplexity=10)
> df.tsne <- data.frame(tsne$Y)
> head(df.tsne)
         X1        X2       X3
1  7.028892 -3.443516 31.23626
2 12.551731  2.492051 23.93481
3 14.325457 -1.469314 23.83501
4 13.171411 -1.026139 22.36516
5  7.846652 -4.932413 31.66133
6  0.824220 -4.808023 35.76660
```

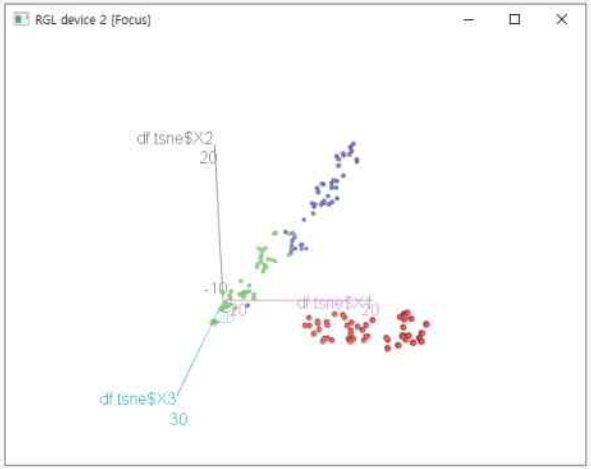
3. 차원 축소

```
> # 회귀면이 포함된 3차원 산점도
> scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3)
>
```



3. 차원 축소

```
> # 회귀면이 없는 3차원 산점도
> points <- as.integer(ds.y)
> color <- c('red','green','blue')
> scatter3d(x=df.tsne$X1, y=df.tsne$X2, z=df.tsne$X3,
+           point.col = color[points],        # 점의 색을 품종별로 다르게
+           surface=FALSE)                    # 회귀면을 표시하지 않음
```



감사합니다.