R computing for Business Data Analytics

Instructor: Dr. Howard Hao-Chun Chuang

Assistant Professor, Department of Management Information Systems

National Chengchi University, Taipei, Taiwan 116

chuang@nccu.edu.tw

Last Revised: October 2014

8.1 Multivariate data, analysis, and visualization

- Introduction

In this lecture you will be exposed to several classical methods for multivariate data analysis. Multivariate analysis is a very broad subject and there is no way we can take a deep dive into the topic within a few hours. My approach is to borrow examples from a nice book by Everitt and Hothorn (2011) and show you various applications of multivariate methods.

Multivariate data arise when people record the values of several outcomes/subjects of interest across a sample of units/objects. Multivariate data are ubiquitous and the examples include:

Chest, waist, and hips measurements on a group of people

The scores for different subjects achieved by students

Carat weight, clarity, color, and cut (4C) for a bunch of diamonds

Take the last case of *diamonds* for example, three questions might be addressed:

Could carat weight, clarity, color, and cut be summarized in some way by combining the four measurements into a single index/number?

Are the subtypes/subgroups of diamonds within which individual diamonds are of similar $4C_s$ and between which $4C_s$ differ?

The first question can be addressed using *principal component analysis* (section 9.2) and the second question can be addressed through *cluster analysis* (section 9.3). Before we start, let's install the following two packages in *R*

```
>install.packages("MVA")
>install.packages("HSAUR2")
>data(USairpollution)
```

We will explore the data "USairpollution" throughout the rest of sections 9.1 & 9.2.

```
>head(USairpollution)
```

SO2: SO_2 content of air in micrograms per cubic metre;

temp: average annual temperature in degrees Fahrenheit

manu: number of manufacturing enterprises employing 20 or more workers

popul: population size (1970 census) in thousands

wind: average annual wind speed in miles per hour

precip: average annual precipitation in inches

predays: average number of days with precipitation per year

- Covariances, correlations, and distances

There are three measures that help us assess the "relationships" between the variables or the relative "closeness" of different units as by their different variable values.

The *covarianc*e of two random variables is a measure of their *linear* dependence.

$$\sigma_{ij} = Cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]$$

If X_i and X_j are independent of each other, their covariance is zero. The converse is NOT true.

If $i=j$, the covariance of the variable with itself is just its variance

$$\sigma_i^2 = E((X_i - \mu_i)^2)$$

In a multivariate data set with q variables, and $q(q\text{-}1)/2$ covariances, which are often arranged in a q by q symmetric matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1q} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q1} & \sigma_{q2} & \cdots & \sigma_q^2 \end{pmatrix}$$

The matrix is generally known as the population *variance-covariance matrix*.

> round(cov(USairpollution), 3) #This is the estimated *sample covariance matrix* **S**

While useful, the covariance is often difficult to interpret because it depends on the scales of the two variables. So, oftentimes people report the *correlation coefficient* instead

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \text{ where } \sigma_i = \sqrt{\sigma_i^2}$$

> round(cor(USairpollution), 3) #This is the estimated *sample correlation matrix* **R**

The last metric is about the *distance* between the units in the data. For two units *i* and *j*, what serves as a measure of distance between them, given the variable values for the two?
The most common measure is *Euclidean distance*, which is critical to cluster analysis (9.2)

$$d_{ij} = \sqrt{\sum_{k=1}^{q} (x_{ik} - x_{jk})^2}$$

where x_{ik} and $x_{jk}, k = 1,...,q$ are the variable values for units *i* and *j*, respectively.

```
> round(dist(scale(USairpollution, center=FALSE)), 3)
```

We use the scale( ) function to divide each variable by its standard deviation since the unit of each variable is quite different.

- The multivariate normal density function

The *multivariate normal distribution* is indispensable to numerous multivariate methods. For a vector of *q* variables, $\mathbf{x}' = (x_1, x_2, ..., x_q)$, the multivariate normal density function is

$$f(\mathbf{x}; \boldsymbol{\mu}; \boldsymbol{\Sigma}) = (2\pi)^{-q/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\}$$

where $\boldsymbol{\mu}$ is the vector of means and $\boldsymbol{\Sigma}$ is the covariance matrix.

The simplest version of the multivariate normal density is the bivariate normal density with *q*=2; this can be written as

$$f((x_1, x_2); (\mu_1, \mu_2), \sigma_1, \sigma_2, \rho) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)} \times \left(\left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 - 2\rho \frac{x_1 - \mu_1}{\sigma_1} \frac{x_2 - \mu_2}{\sigma_2} + \left(\frac{x_2 - \mu_2}{\sigma_2}\right)^2 \right)\right\}$$

The correlation parameter ρ captures the stochastic dependencies between the two normal random variables. This idea of dependencies is super important for risk analysis in finance, insurance, agriculture, etc. If $\rho = 0$, we will have two independent normal distributions.

```
> mu1=0
> mu2=0.5
> sig1=0.5
> sig2=2
> rho=0.5
```

```
> x=seq(-3, 3, 0.01)
> y=seq(-3, 3, 0.01)
> install.packages("fMultivar")
> bivariate=function(x, y){
term1=1/(2*pi*sig1*sig2*sqrt(1-rho^2))
term2=(x-mu1)^2/sig1^2
term3= -(2*rho*(x-mu1)*(y-mu2))/(sig1*sig2)
term4= (y-mu)^2/sig2^2
z=term2+term3+term4
term5=term1*exp((-z/(2*(1-rho^2))))
return(term5)
}
> z=outer(x, y, bivariate)
> persp(x, y, z, main= "Bivariate Normal Distribution", , theta = 55, phi = 30, r = 40,
d = 0.1, expand = 0.5, ltheta = 90, lphi = 180, shade = 0.4, ticktype = "detailed", nticks=5)
```

8.2 Principal component analysis

In the domain of optimization/analytics, the problem of having too many variables is known as the "curse of dimensionality", which brings us to principal component analysis (PCA), a super popular and powerful technique with a central aim – reducing the dimensionality of a multivariate data set, while accounting for as much of the original variation in the data set (or minimizing information loss in dimension-reduction). PCA is often used by economists who need to summarize commodity prices, wage rates, cost of living, etc. into a single index.

The basic goal of PCA is to describe variation in a set of *correlated variables* $\mathbf{x} = (x_1, ..., x_q)$, in terms of a new set of *uncorrelated variables* $\mathbf{y} = (y_1, ..., y_q)$, each of which is a linear combination of the **x** variables. The variables $\mathbf{y} = (y_1, ..., y_q)$ are the **principal components**. They are derived in decreasing order of "importance" in the sense that y_1 accounts for as much as possible of the variation in the original data among all linear combinations of **x**. The y_2 is chosen to account for as much as possible of the remaining variation, subject to being uncorrelated with y_1, and so on.

The general hope of performing PCA is that the first few (<<q) components will account for a substantial proportion of the variation in the original variables $\mathbf{x} = (x_1, ..., x_q)$. Consequently, the principal components $\mathbf{y} = (y_1, ..., y_q)$ provide a succinct summary of **x**. Also, the newly identified components can serve as inputs to some other analyses, such as regression analysis, or cluster analysis to be introduced soon.

The technical details of how to find principal components involve constrained optimization and knowledge of matrix algebra. To avoid confusion, I will skip the math behind PCA and just show you the application of PCA. Nonetheless, remember that PCA is useful when

There are too many explanatory variables relative to the number of observations

The explanatory variables are highly correlated

Go back to the *USairpollution* data. Before running PCA, let's visualize the data we have. The first graph is a *scatterplot matrix*

```
> pairs(USairpollution, pch=".", cex=2)
```

We can do plotting in various 3-dimensional ways

```
> library(scatterplot3d)
> s3d=scatterplot3d(temp, wind, SO2, type= "h", angle=55)
> fit=lm(SO2~temp+wind)
> s3d$plane3d(fit)
```

The plot could even be interactive/spinnable

```
> library(rgl)
> plot3d(temp, wind, SO2, col='red', size=3)
```

Alternatively

```
> library(Rcdmr)
> scatter3d(temp, wind, SO2)
```

Finally, we can generate some more sophisticated conditional scatterplots

```
> library(lattice)
> plot(xyplot(SO2 ~ temp| cut(wind, 2)))
> plot(xyplot(SO2 ~ temp| cut(wind, 3), layout=c(3,1))
> pollution=with(USairpollution, equal.count(SO2, 4))
> plot(cloud(precip ~ temp*wind | pollution, panel.aspect=0.9))
```

Now switch to PCA in *R*. Since the SO2 is the response variable, we want to concentrate on the other predictor variables, two of which relate to human ecology (popul, manu) and four to climate (temp, wind, precip, predays).

```
> round(cor(USairpollution[ , -1]), 3)
> usair_pca=princomp(scale(USairpollution[ , -1]))
> summary(usair_pca, loadings=TRUE)
```

So, how many principal components do we need? Let's look at the *scree diagram*

```
> plot(usair_pca$sdev^2, xlab= "component number", ylab= "eigenvalue", type='l')
> abline(h=1, lty=2, col= 'red')
```

How do we calculate scores of the new variables?

```
> usair_pca$scores[ , 1]
> usair_pca$loadings[ , 1]%*%t(scale(USairpollution)[,-1])
```

We can further use the scores in regression analysis

```
> usair_reg1=lm(SO2 ~ ., data=USairpollution)
> usair_reg2=lm(SO2 ~ usair_pca$scores, data=USairpollution)
> summary(usair_reg1)
> summary(usair_reg2)
> usair_reg3=lm(SO2 ~ usair_pca$scores [ ,c(1, 4:6) ], data=USairpollution)
```

Notice that, the components with small variance do not necessarily have small correlations with the response variable SO2. Why is that?

8.3 Cluster analysis

One of the natural tendencies of living creatures involves the grouping of similar objects. For instance, animals are called cats, dogs, horses, etc., and each name/label collects individuals into groups. When it comes to data, it is very common that the analyst is interested in finding a classification in which the items of interest are sorted into a small number of *homogeneous groups* or *clusters*. In most cases, the classified clusters are required to be *mutually exclusive* rather than *overlapping*.

Cluster analysis is a generic term for different numerical methods with the common objective of uncovering groups of observations that are homogeneous and separated from other groups. The major outcome is a set of *class labels* that provide a parsimonious way of describing the

patterns of intra-group similarities and inter-group differences. For example, based on values of various **economic metrics**, countries could be labelled as *developing* or *developed* ones. That said, the same countries can be labelled as *popular*, or *unpopular* ones based on values of various **tourism metrics**. The point is that, a variety of clusters will always be possible for whatever is being grouped, depending on what kind of metrics/angles you are using. In this section, we will briefly discuss two classical and oft-used clustering methods: *agglomerative hierarchical clustering* and *k-means clustering*.

Remember in section 9.1 we talked about the distance between units *i* and *j*:

$$d_{ij} = \sqrt{\sum_{k=1}^{q} (x_{ik} - x_{jk})^2}$$

Based on this inter-individual distance measure, the hierarchical clustering uses three inter-group measures for two clusters *A* and *B*

Single linkage: $d_{AB} = \min_{i \in A,\, j \in B} (d_{ij})$

Complete linkage: $d_{AB} = \max_{i \in A,\, j \in B} (d_{ij})$

Group average: $d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$

Let's check it out in *R*.

```
> dm=dist(scale(USairpollution[ , -1]))
> dev.new(width=15, height=5)
> par(mfrow=c(1,3))
> plot(cs <- hclust(dm, method= "single"))
> plot(cc <- hclust(dm, method= "complete"))
> plot(ca <- hclust(dm, method= "average"))
```

We can further define the number of clusters

```
> cutree(cs, h=2)
> table(cutree(cs, h=2))
```

We could perform the cluster analysis based on the earlier identified principal components

```
> dm_pc=dist(usair_pca$scores[ , 1:3])
> plot(cs_pc <- hclust(dm_pc, method= "single"))
> table(cutree(cs_pc, h=2))
```

Are the results different?

Now, how about removing Chicago and Phoenix?

I want to wrap up the section talking about the *k*-means clustering technique, which seeks to partition the *n* individuals into *k* groups or clusters $(G_1,...G_k)$, where G_i denotes the set of n_i individuals in the i_{th} group, for a GIVEN *k*. In most cases people try to find the partition of the *n* individuals to *k* groups that minimize the within-group sum of squares (WGSS) over all variables

$$WGSS = \sum_{j=1}^{q}\sum_{\ell=1}^{k}\sum_{i\in G_\ell}(x_{ij} - \bar{x}_j^{(\ell)})^2$$

where $\bar{x}_j^{(\ell)} = \frac{1}{n_i}\sum_{i\in G_\ell} x_{ij}$ is the mean of the individual in group G_ℓ on variable *j*.

This seemingly intuitive objective function, however, is computationally very extensive. The number of possible partitions for sample size *n* and clusters *k* is

$$\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$$

which is a *Stirling number of the second kind*. For a very small problem of grouping *n*=15 individuals into *k*=3 clusters, the number of possible partitions is 2,375,101 (verify it in *R*!). So, there is no way to enumerate every possible partition even using the best computer in the world. Picking the optimal number of *k* and the best algorithm/heuristic for partitioning is beyond the scope of this course. Here we focus on how to do conduct the analysis in *R*.

```
> help(kmeans)
> usair_ck= kmeans(scale(USairpollution[ , -1]), centers=3))
> usair_ck$cluster
```

The results are somewhat different from hierarchical clustering, aren't they?

Let's explore the best number of clusters *k* a bit.

```
> k=6
> WGSS=c()
> for(i in 1:k){
>     WGSS[i]=sum(kmeans(scale(USairpollution[ , -1]), centers=i)$withinss)
> }
```

```
> plot(1:k, WGSS, type= “b”)
```

What is the number *k* we may want to stick to?

Think about the following example before we finish. One can cluster books based on subject matter into classes such as dictionaries, novels, biographies, and so on. Instead, he/she could cluster books based on the color of the book’s binding, the number of pages, etc. Apparently, the former clustering results are going to be more useful. So, your job as a data analyst is to identify relevant metrics for your problem domain such that your findings from cluster analysis will have practical value. Keep it in mind that, in the end of day, any clustering of objects is likely to be judged on its USEFULNESS.