

# Unsupervised Learning: Dimensionality Reduction

Dr Liew How Hui

Jan 2021

# Reminder

- Submission of **Report + Programming Code**:
  - ▶ Week 11 Wednesday.
  - ▶ In proper formats, marks will be deducted for improper formats.
  - ▶ By Email from Group Leader.
- Oral Presentation (Week 11 to Week 13?):
  - ▶ Arrangement: In reverse order of registration — the latest to register will present first, the first group will present last or have the right to choose time slot.
  - ▶ Presentation: 18–25 minutes? In class recording.
  - ▶ Submission of **presentation slides**.

13 groups 1 week 5–6 groups?

# Outline

1 Overview

2 PCA

3 SNE & t-SNE

# Unsupervised Learning

When the data we have does not have any labels, it is impossible for us to apply supervised learning.

E.g. The DNA or RNA data of virus is such an example.

Unsupervised learning is often performed as part of an *pattern recognition*.

It is difficult to assess the results obtained from unsupervised learning methods → no “indisputable” performance metric!

# Unsupervised Learning

For an unlabelled  $n \times p$  **numeric** data, it may have the following patterns:

- $n$  is much larger than  $p$  and the data is distributed along some subspace in  $\mathbb{R}^p \rightarrow$  dimension reduction
- $n$  is small and  $p$  is large, the data are correlated and the features are linear combinations of fewer features  $\rightarrow$  dimension reduction
- Data are in closely-linked nonlinear distributed regions in high-dimensional feature space  $\rightarrow$  nonlinear dimension reduction
- Data are in “clusters”  $\rightarrow$  clustering (next topic)

# Dimensionality Reduction

Target: unlabelled data  $X$  with a set of **numeric features**  $X_1, \dots, X_p$  measured on  $n$  observations

“Dimensionality reductions” are a class of unsupervised learning methods which try to **extract** essential “dimension” information of the data to “characterise” the data.

Some dimensionality reduction algorithms that are PCA (linear), SNE (nonlinear), t-SNE (non-parametric/nonlinear), Sammon mapping (nonlinear), Isomap (nonlinear), LLE (nonlinear), CCA (Canonical-Correlation Analysis, nonlinear), MVU (Maximum Variance Unfolding, nonlinear), Laplacian Eigenmaps (nonlinear), etc.

# Dimensionality Reduction (cont)

The good news is that we need to study only PCA and t-SNE (t-Distributed Stochastic Neighbour Embedding) to effectively visualise data in lower dimensions.

The PCA tries to find a global structure and this can lead to local inconsistencies, i.e. far away point can become nearest neighbours.

The t-SNE tries to preserve local structure, i.e. the low dimensional neighbourhood “representation” should be the same as original neighbourhood.

# Outline

1 Overview

2 PCA

3 SNE & t-SNE



# PCA

Given unlabelled data  $X$  of shape  $n \times p$ .

Motivations:

If one of the column in  $X$  is constant (e.g. 0), can we remove it?  $\rightarrow$  remove “redundant” data!

What is the statistical concept that measures the correlation between columns in statistics?

# PCA (cont)

Answer: Covariance matrix of  $X$ .

**Step 0:** Check the variance of each columns and decide if scaling is necessary.

**Step 1:** Shift the data to “column mean” / centre  $\bar{x}$ ,

$$x_{.j} \rightarrow x_{.j} - \bar{x}_j =: \tilde{X}$$

**Step 2 (optional):** If scaling is necessary,

$$s_{.j} = \sqrt{\frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}{n-1}} \Rightarrow x_{.j} \rightarrow \frac{x_{.j} - \bar{x}_j}{s_{.j}} =: \tilde{X}$$

Note: In exam, don't standardise with Step 2 unless it is stated.

# PCA (cont)

**Step 3:** Compute the covariance matrix of  $X$ :

$$\Sigma := \text{Cov}(X) = \frac{1}{n-1} \tilde{X}^T \tilde{X}$$

**Step 4:** Compute the eigenvalues and eigenvectors of  $\Sigma$ .

But Why?

Answer: <https://scikit-learn.org/stable/modules/decomposition.html>  $\rightarrow$  decomposing the matrix  $\Sigma$  using linear algebra's "spectral theorem".

## PCA (cont)

A matrix  $\Sigma$  is said to be *non-negative definite* if

$$\mathbf{x}\Sigma\mathbf{x}^T \geq 0, \quad \forall \mathbf{x} = (x_1, \dots, x_p), \quad x_1^2 + \dots + x_p^2 = 1.$$

Spectral Theorem: A non-negative definite matrix can be decomposed into

$$\Sigma = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \dots + \lambda_q \mathbf{v}_q \mathbf{v}_q^T, \quad q \leq \min\{n, p\}. \quad (1)$$

Here,  $\lambda_i$  are the “eigenvalues” and  $\mathbf{v}_i$  are the “eigenvectors”. The eigenvectors are orthogonal to each other and  $\mathbf{v}_i^T \mathbf{v}_i = 1$ . Note that

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_q.$$

# PCA (cont)

**Step 5** (optional): Construct the principal components:

$$PC_1 = e_{11}\tilde{X}_1 + e_{12}\tilde{X}_2 + \cdots + e_{1q}\tilde{X}_q,$$

$$PC_2 = e_{21}\tilde{X}_1 + e_{22}\tilde{X}_2 + \cdots + e_{2q}\tilde{X}_q,$$

$$\vdots$$

$$PC_q = e_{q1}\tilde{X}_1 + e_{q2}\tilde{X}_2 + \cdots + e_{qq}\tilde{X}_q.$$

where  $v_i = (e_{i1}, e_{i2}, \dots, e_{iq})$ .

# PCA (cont)

## Example 11.1.1

Given an unlabelled data with 15 observations and feature columns (but tabulate in horizontal form to save space). Find all the principal components.

	1	2	3	4	5	6	7	8
$X_1$	8.095	6.91	4.119	4.4	4.65	2.329	8.272	4.595
$X_2$	4.104	5.272	4.063	5.366	5.238	4.711	2.46	0.581
$X_3$	2.351	-2.827	-3.786	-0.261	-1.096	-1.456	-1.727	-1.292

	9	10	11	12	13	14	15
$X_1$	8.071	6.403	4.136	7.283	2.744	4.939	4.924
$X_2$	5.883	3.624	3.514	-1.301	3.584	3.024	2.754
$X_3$	0.938	2.949	2.918	-0.738	3.866	-0.803	5.154

# PCA (cont)

## Example 11.1.1 (cont)

**Solution:** It is possible for us to find the principal components using linear algebra. First, let us, calculate the column means:

$$\bar{x}_{\cdot j} = (5.45800, 3.52513, 0.27933)$$

and shift  $X$  to the “centre”, i.e.  $\tilde{X} = x_{\cdot j} - \bar{x}_{\cdot j}$ :

	1	2	3	4	5	6	7	8
$X_1$	2.637	1.452	-1.339	-1.058	-0.808	-3.129	2.814	-0.863
$X_2$	0.578867	1.746867	0.537867	1.840867	1.712867	1.185867	-1.065133	-2.944133
$X_3$	2.071667	-3.106333	-4.065333	-0.540333	-1.375333	-1.735333	-2.006333	-1.571333

	9	10	11	12	13	14	15
$X_1$	2.613	0.945	-1.322	1.825	-2.714	-0.519	-0.534
$X_2$	2.357867	0.098867	-0.011133	-4.826133	0.058867	-0.501133	-0.771133
$X_3$	0.658667	2.669667	2.638667	-1.017333	3.586667	-1.082333	4.874667

# PCA (cont)

## Example 11.1.1 (cont)

The covariance matrix of  $X$  is

$$\text{Cov}(X) = \frac{1}{15-1} \tilde{X}^T \tilde{X} = \begin{bmatrix} 3.700046 & -0.441594 & -0.261248 \\ -0.441594 & 3.63697 & -0.097637 \\ -0.261248 & -0.097637 & 6.82242 \end{bmatrix}$$

We need to find the eigenvalues of  $\text{Cov}(X)$ . This can be done using university linear algebra (determinant, r.e.f)

$$\lambda_i = 6.845301, 4.105652, 3.208484$$

Using R: `eigen(S)`.



# PCA (cont)

## Example 11.1.1 (cont)

The corresponding eigenvectors are the **weighted vectors**

$$\mathbf{v}_1 = \begin{bmatrix} -0.080068 \\ -0.019308 \\ 0.996602 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0.722438 \\ -0.689991 \\ 0.044673 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 0.686784 \\ 0.723560 \\ 0.069195 \end{bmatrix}$$

The first, second and third principal components of  $X$  are:

$$PC_1 = -0.080068X_1 - 0.019308X_2 + 0.996602X_3$$

$$PC_2 = 0.722438X_1 - 0.689991X_2 + 0.044673X_3$$

$$PC_3 = 0.686784X_1 + 0.723560X_2 + 0.069195X_3$$

# PCA (cont)

## Example 11.1.1 (cont)

### Python

```
pca = sklearn.decomposition.PCA(); pca.fit(X)
```

### R

```
pca = prcomp(X[, scale=TRUE])
```

pca\$center:  $\bar{x}_{.j}$

pca\$sdev:  $\sqrt{\lambda_i}$

pca\$rotation:  $[v_1, v_2, \dots]$

pca\$x:  $[\tilde{X}v_1, \tilde{X}v_2, \dots]$

---

pca\$scale:  $\sqrt{\frac{\sum_i (x_{ij} - \bar{x}_{.j})^2}{n-1}}$

# PCA (cont)

Final Exam Jan 2019, Q1(c); Need to use “linear algebra”

---

You are given the following information:

- The data set consists of 3000 observations and 2 predictors,  $X_1$  and  $X_2$ .
- The covariance matrix,  $C$  of  $X_1$  and  $X_2$  is

$$C = \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix}$$

# PCA (cont)

## Final Exam Jan 2019, Q1(c) cont.

- (i) Compute the eigenvalues,  $\lambda_1$  and  $\lambda_2$ . For each of the eigenvalues computed, find the eigenvectors,  $v_1$  and  $v_2$ . (7 marks)
- (ii) First principal component, PC1 is the linear combination of predictors that has the maximum variance among all principal components. Based on your answer in (i), write the equation of first principal component for this data set. (2 marks)

# PCA (cont)

Solution:

(i) Eigenvalues:

$$\begin{aligned}|C - \lambda I| &= \begin{vmatrix} 2.0 - \lambda & 0.8 \\ 0.8 & 0.6 - \lambda \end{vmatrix} \\ &= \lambda^2 - 2.6\lambda + 0.56 = 0 \\ \Rightarrow x &= 0.236985, 2.363015\end{aligned}$$

[2 marks]

# PCA (cont)

Solution (cont):

Eigenvectors:

$\lambda = 0.236985$ :

$$\begin{bmatrix} 2.0 - 0.236985 & 0.8 \\ 0.8 & 0.6 - 0.236985 \end{bmatrix} \mathbf{v}_1 = 0 \Rightarrow$$

$$\mathbf{v}_1 = \frac{1}{\sqrt{0.8^2 + (-1.763015)^2}} \begin{bmatrix} 0.8 \\ -1.763015 \end{bmatrix} = \begin{bmatrix} 0.413216 \\ -0.910633 \end{bmatrix}$$

[2.5 marks]

## PCA (cont)

$\lambda = 2.363015$ :

$$\begin{aligned} & \begin{bmatrix} 2.0 - 2.363015 & 0.8 \\ 0.8 & 0.6 - 2.363015 \end{bmatrix} \mathbf{v}_2 = \mathbf{0} \\ \Rightarrow \mathbf{v}_2 &= \frac{1}{\sqrt{0.8^2 + (-(-0.363015))^2}} \begin{bmatrix} 0.8 \\ -(-0.363015) \end{bmatrix} \\ &= \begin{bmatrix} 0.910633 \\ 0.413217 \end{bmatrix} \quad [2.5 \text{ marks}] \end{aligned}$$

(ii) PC1 is the linear combination of predictors with highest eigenvalue  $\lambda_1 = 2.363015$ , i.e.

$$PC_1 = 0.910633X_1 + 0.413217X_2.$$

# PCA (cont)

A little bit more theory: Let  $Y = \frac{1}{\sqrt{n-1}}\tilde{X}$  and  $\lambda_i = \sigma_i^2$ .  
From (1), we have

$$Y^T Y = \sigma_1^2 \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \sigma_q^2 \mathbf{v}_q \mathbf{v}_q^T$$

Since  $\mathbf{v}_i^T \mathbf{v}_j = 0$  for  $i \neq j$ ,  $\mathbf{v}_i^T \mathbf{v}_i = 1$ , we have

$$\begin{aligned} Y^T Y \mathbf{v}_i &= (\sigma_1^2 \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \sigma_q^2 \mathbf{v}_q \mathbf{v}_q^T) \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i \\ \Rightarrow Y^T \left( \frac{1}{\sigma_i} Y \mathbf{v}_i \right) &= \sigma_i \mathbf{v}_i. \end{aligned}$$



## PCA (cont)

Let  $u_i = \frac{1}{\sigma_i} Y v_i$  and  $U = [u_1, \dots, u_q]$ . Putting left and right column vectors into matrix form, we have

$$Y^T U = VS, \quad V = [v_1, \dots, v_q], S = \text{diag}(\sigma_1, \dots, \sigma_q).$$

From which we obtain the singular value decomposition (SVD):

$$Y^T = VSU^T \Rightarrow \boxed{Y = USV^T}.$$

Note that  $S^T = S$ ,  $U$  is related to `pca$x`,  $V$  is related to `pca$rotation` and  $\sigma_i$  is `pca$sdev` mentioned earlier in Example 11.1.1.

## PCA (cont)

The proportion of variation explained by the  $k$ th principal component is defined to be the eigenvalue,  $\lambda_k$  for that component divided by the sum of the eigenvalues. In other words, the  $k$ th principal component explains the following proportion of the total variation, which denote as proportion of variance,  $PVE_k$ , explained by  $PC_k$ :

$$PVE_k = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}, \quad k = 1, \cdots, p. \quad (2)$$

## PCA (cont)

A related quantity is the proportion of variation explained by the first  $k$  principal components is the cumulative proportion of variance explained until  $PC_k$ :

$$CPVE_k = \frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}, \quad k = 1, 2, \cdots, p. \quad (3)$$

In another word,  $CPVE_k$  is the cummulation of  $PVEs$ .

## PCA (cont)

If the proportion of variation explained by the first  $k$  principal components is large, then not much information is lost by considering only the first  $k$  principal components. In general, we only retain the first  $k$  principal components and balance the two conflicting desires:

- 1 To obtain the simplest possible interpretation, **we want  $k$  to be as small as possible.**
- 2 To avoid loss of information, we want

$$CPVE_k \approx 1.$$

# PCA (cont)

How many principal components should we use? One of the methods to determine the number of principal components is to look at a *scree plot* ([https://en.wikipedia.org/wiki/Scree\\_plot](https://en.wikipedia.org/wiki/Scree_plot)). With the eigenvalues ordered from largest to the smallest, a scree plot is the plot of “principal component”  $\lambda_k$  against  $k$ . The number of components is determined at the point beyond which the remaining eigenvalues are all relatively small and of comparable size.

# PCA (cont)

## Example 11.1.3

The dataset (<https://www.openml.org/d/509>) from the guide “Places Rated Almanac”, by Richard Boyer and David Savageau, copyrighted and published by Rand McNally, uses nine rating criteria: (1) Climate and Terrain, (2) Housing, (3) Health Care & Environment, (4) Crime, (5) Transportation, (6) Education, (7) The Arts, (8) Recreation, (9) Economics.

Note that within the dataset, except for housing and crime, the higher the score the better. For housing and crime, the lower the score the better. Where some communities might rate better in the arts, other communities might rate better in other areas such as having a lower crime rate and good educational opportunities.

# PCA (cont)

## Example 11.1.3 (cont.)

Suppose that principal component analysis has been implemented and the eigenvalues of each principal component are shown in the table below.

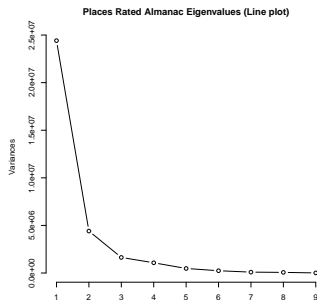
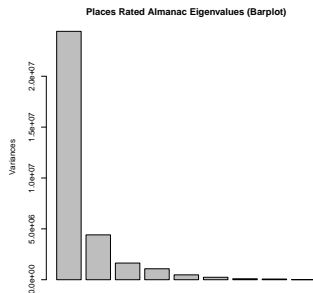
$k$	1	2	3	4	5	6
$\lambda_k$	24413669	4408005	1638040	1076356	478338.3	240851.8
$k$	7	8	9			
$\lambda_k$	92809.94	66995.9	10962.63			

Compute the proportion of variance explained by each component and the cumulative proportion respectively. After that, construct a scree plot and determine the number of principal components to be considered.

# PCA (cont)

## Example 11.1.3 (cont.)

There are two ways to represent “scree plot”, one using “bar chart”, the other using “line plot”.





# PCA (cont)

## Example 11.1.4

The correlations between the principal components and the original variables for Example 11.1.3 are shown in the table below.

Variable	PC1	PC2	PC3
Climate	0.00642	-0.01546	0.00669
Housing	0.26914	-0.93721	0.08264
Health	0.17832	0.02054	-0.02776
Crime	0.02813	0.01090	-0.03761
Transportation	0.14930	-0.01876	-0.97153
Education	0.02519	0.00140	-0.04151
Arts	0.93086	0.28226	0.15103
Recreation	0.06982	-0.10385	-0.14957
Economy	0.02513	-0.17336	-0.01274

# PCA (cont)

## Example 11.1.4 (cont)

### **Solution:**

#### First Principal Component - PC1

The first principal component is **strongly correlated** with the original variable Arts. It increases with all features, in particular, Arts. Based on the correlation of 0.93086, the first principal component is primarily a measure of the Arts and it has  $0.93086 \times PVE_1 = 70\%$  influence. It would follow that communities with high values tend to have a lot of arts available, in terms of museums, fine arts, public radio stations, public television stations, universities offering degrees in the arts, symphony orchestras, theatres, opera companies, dance companies, public libraries etc. Whereas communities with small values would have very few of these types of opportunities.

# PCA (cont)

## Example 11.1.4 (cont)

### Second Principal Component - PC2

The second principal component decreases with the feature Housing and increases with the feature Arts. This component can be viewed as a measure of things utility bills, property taxes, mortgage payments related to Housing have 12.7% ( $0.93721 \times PVE_2$ ) significance.

# PCA (cont)

## Example 11.1.4 (cont)

### Third Principal Component - PC3

The third principal component decreases mainly with increasing Transportation and increases slightly with increasing Arts. This suggests that places with bad transportation has 4.9% ( $0.97153 \times PVE_3$ ) significance.

# PCA (cont)

## Final Exam May 2019, Q1(c)

An investigation is carried out to examine the eating habit of citizens in each state in Malaysia. A total of 8 types of food had been investigated — the average consumption of each type of food (grams) per person per week in each state in Malaysia was recorded. The 8 types of food are listed below:

- $x_1$  = Fish
- $x_2$  = Meat
- $x_3$  = Grain
- $x_4$  = Dairy
- $x_5$  = Beans
- $x_6$  = Egg
- $x_7$  = Vegetables
- $x_8$  = Fruit

# PCA (cont)

## Final Exam May 2019, Q1(c) cont

For advanced analysis on the eating habit, the analyst would like to reduce the dimension of the data.

Therefore, principal component analysis has been performed. Eigenvalues computed from the principal component analysis are

$$\lambda = [0.0342, 0.6432, 2.3664, 0.5869, 1.1894, 0.0032, 5.6379, 0.0179]^T$$

# PCA (cont)

## Final Exam May 2019, Q1(c) cont

- ① State the variance explained by each principal component. With a targeted cumulative proportion of variance explained (CPVE) of 85%, state the number of principal components to be considered. (4 marks)
- ② Given that the eigenvector for PC1 is

$$\mathbf{e}_1 = [0.174 \quad -0.626 \quad -0.146 \quad 0.667 \quad -0.249 \quad 0.743 \quad 0.598 \quad 0.484] .$$

Interpret the principal component results with respect to the types of food. You should provide explanation on which types of food are the most and the least contributed to PC1, as well as the correlation between PC1 and that variable. (5 marks)

# PCA (cont)

## Solution: (i)

Variance explained = eigenvalue (Highest for PC1 in descending order)

PC	$\lambda$	PVE	CPVE
1	5.6379	0.5380	0.5380
2	2.3664	0.2258	0.7638
3	1.1894	0.1135	0.8773
4	0.6432	0.0614	0.9387
5	0.5869	0.0560	0.9947
6	0.0342	0.0033	0.9980
7	0.0179	0.0017	0.9997
8	0.0032	0.0003	1.0000
Total	10.4791		

With targeted CPVE of 85%, 3 principal components (PC1, PC2 and PC3), with CPVE of 87.73%, should be considered.



# PCA (cont)

## Solution: (ii)

The correlation between PC1 and variables are shown below:

Variable, $x_i$	Type of Food	$e_{1i}$
$x_1$	Fish	0.174
$x_2$	Meat	-0.626
$x_3$	Grain	-0.146
$x_4$	Dairy	0.667
$x_5$	Beans	-0.249
$x_6$	Egg	0.743
$x_7$	Vegetables	0.598
$x_8$	Fruit	0.484

# PCA (cont)

## **Solution:** (ii)

Egg is the most contributed to PC1 as it has the highest absolute in the eigenvalues which is 0.743. Egg is positively correlated with PC1 since it has a positive value. This means that PC1 will be higher if the egg consumption increases.

Grain is the least contributed to PC1 as it has the lowest absolute in the eigenvalues which is  $-0.146$ . Grain is negatively correlated with PC1 since it has a negative value. This means that PC1 will be lower if the grain consumption increases.

# PCA (cont)

A *biplot* is a 2D “scatter plot” of  $PC_1$  and  $PC_2$

In the above, we required that  $n > p$ . However, for the case of  $n \leq p$ , we just need to transpose the data  $X$  and all the decomposition can be carried out.

For the case  $n \leq p$ , it corresponds to data that are collected on a **large number of variables** from a single population.

There would be too many pairwise correlations between the variables to consider, i.e. there are  $\binom{p}{2} = \frac{p(p-1)}{2}$  correlations or scatter plots to be examined. E.g. when  $p = 10$ , there are 45 scatter plots!

# PCA (cont)

A *biplot* is a 2D “scatter plot” of the first and the second principle components. In the above, we required that  $n > p$ . However, for the case of  $n \leq p$ , we just need to transpose the data  $X$  and all the decomposition can be carried out.

For the case  $n \leq p$ , it corresponds to data that are collected on a **large number of variables** from a single population. With a large number of variables, the dispersion matrix may be too large to study and interpret properly. There would be too many pairwise correlations between the variables to consider, i.e. there are  $\binom{p}{2} = \frac{p(p-1)}{2}$  correlations or scatter plots to be examined. For example, when  $p = 10$ , there are 45 scatter plots! Hence, graphical displays may also not be particularly helpful when the data set is very large.

# PCA (cont)

## Example 11.1.7

Consider the “EATING IN THE UK” from <http://setosa.io/ev/principal-component-analysis/>. This is a **17-dimension** example. The table below is the average consumption of 17 types of food in grams per person per week for every country in the UK.

# PCA (cont)

## Example 11.1.7 (cont)

	England	Wales	Scotland	N.Ireland
Cheese	105	103	103	66
Carcass meat	245	227	242	267
Other meat	685	803	750	586
Fish	147	160	122	93
Fresh potatoes	720	874	566	1033
Processed potatoes	198	203	220	187
Fats and oils	193	235	184	209
Sugars	156	175	147	139
Fresh Veg	253	265	171	143
Other Veg	488	570	418	355
Processed Veg	360	365	337	334
Fresh fruit	1102	1137	957	674
Cereals	1472	1582	1462	1494
Beverages	57	73	53	47
Soft drinks	1374	1256	1572	1506
Alcoholic drinks	375	475	458	135
Confectionery	54	64	62	41

# PCA (cont)

## Example 11.1.7 (cont)

The PVEs and CPVEs are given as a summary to PCA:

---

Importance of components:

	PC1	PC2	PC3	PC4
Standard deviation	324.1502	212.7478	73.87622	4.189e-14
Proportion of Variance	0.6744	0.2905	0.03503	0.000e+00
Cumulative Proportion	0.6744	0.9650	1.00000	1.000e+00

---

# PCA (cont)

## Example 11.1.7 (cont)

A script to compute the above values and produce a biplot is listed below.

---

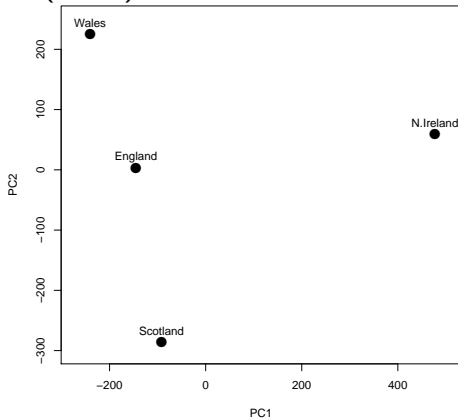
```
1 # From https://bioboot.github.io/bggn213_f17/class-material/UK_food_p
2 x = read.csv("UK_foods.csv")
3 rownames(x) = x[,1]
4 x = x[,-1]          # Remove first column
5 ### Print x as a nice table (an alternative is R's View() function)
6 #knitr::kable(x, caption="The full UK foods data table")
7 pca = prcomp(t(x))
8 print(pca$sdev^2)
9 print(summary(pca))
0 plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500),
1      ylim=c(-300,250), pch=16, cex=2)
2 text(pca$x[,1], pca$x[,2]+19, colnames(x))
3 # biplot(pca)
```

---



# PCA (cont)

## Example 11.1.7 (cont)



# PCA (cont)

## Example 11.1.7 (cont)

From the biplot, we detect that Northern Ireland is a major outlier. Once we go back and look at the data in the table, this makes sense: the Northern Irish eat way more grams of fresh potatoes and way fewer of fresh fruits, cheese, fish and alcoholic drinks. It's a good sign that structure we've visualised reflects a major fact of real-world geography: Northern Ireland is the only of the four countries not on the island of Great Britain.

# Outline

1 Overview

2 PCA

3 SNE & t-SNE

# SNE and t-SNE

PCA is a linear algorithm and it cannot “project” the nonlinear relationship between features to low dimensional space well. On the other hand, *SNE* (*Stochastic Neighbour Embedding*) and *t-SNE* (*t-Distributed SNE*) use probability distributions with random walk on neighbourhood graphs to identify and try to preserve the “nonlinear” structure of the data.

# SNE and t-SNE (cont)

SNE converts the high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities, i.e.

$$\mathbb{P}(X = x_j | X = x_i) = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2})} =: p_{j|i}$$

where  $\sigma_i$  is the variance of the Gaussian that is centred on the data point  $x_i$ .

# SNE and t-SNE (cont)

Assume that the “projected” high-dimensional points  $x_i$  and  $x_j$  to a low-dimension “space” is  $y_i$  and  $y_j$  respectively. The conditional probability of  $y_i$  and  $y_j$  is

$$\mathbb{P}(Y = y_j | Y = y_i) = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)} =: q_{j|i}. \quad (4)$$

Note that, we define

$$\mathbb{P}(Y = y_i | Y = y_i) = 0, \quad \text{for all } i$$

since we only want to model pair-wise similarity.

# SNE and t-SNE (cont)

To measure the minimisation of sum of difference of conditional probability, SNE minimises the sum of *Kullback-Leibler divergences* (mentioned in the Entropy Section) overall data points using a gradient descent method:

$$C = \sum_i \sum_j p_{j|i} \ln \frac{p_{j|i}}{q_{j|i}}. \quad (5)$$

In other words, the SNE cost function (5) focuses on retaining the local structure of the data in the map (for reasonable values of the variance of the Gaussian in the high-dimensional space,  $\sigma_i$ ).

# SNE and t-SNE (cont)

t-SNE tries to minimise symmetric version of the SNE cost function:

$$C = \sum_i \sum_j p_{ij} \ln \frac{p_{ij}}{q_{ij}}, \quad p_{ij} := \frac{p_{j|i} + p_{i|j}}{2n}, \quad q_{ij} := \frac{q_{j|i} + q_{i|j}}{2n}$$

and employs a **heavy-tailed distribution in the low-dimensional space**:

$$\mathbb{P}(Y = y_j | Y = y_i) = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2)^{-1}} =: q_{j|i} \quad (6)$$

to alleviate both the crowding problem & the optimisation problems of SNE.



# SNE and t-SNE (cont)

In R, t-SNE is implemented in `tsne::tsne` and `Rtsne::Rtsne` (based on C++, much faster). In Python, t-SNE is implemented as `sklearn.manifold.TSNE`. Therefore many other manifold unsupervised learning methods in `sklearn.manifold` which are not explored, such as `Isomap`, `LocallyLinearEmbedding` (contain a lot of variance: `method='standard', 'hessian', 'modified' or 'ltsa'`), `SpectralEmbedding`, `MDS` (multi-dimensional scaling), etc.

# SNE and t-SNE (cont)

Example 11.2.1: An R script to compare PCA and t-SNE on the MNIST data is listed below.

```
1 # https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r
2 library(Rtsne) # Uses Barnes-Hut-TSNE algorithm instead of the slower
3 train = read.csv("mnist_train.csv")
4 X = train[,-1]
5 train$label = as.factor(train$label)
6 colours = rainbow(length(unique(train$label)))
7 names(colours) = unique(train$label)
8
9 # https://www.youtube.com/watch?v=xPB0-MMxIoQ[R Tutorial: PCA and t-S
0 pca = prcomp(train[,-1], rank=2) # project data to first two PCs only
1 plot(pca$x[,1:2], pch=as.character(train$label),
2      col=colours[train$label], main="Biplot")
3
4 # Takes a long time to calculate: dim(X) = 10000 x 784
5 tsne = Rtsne(X, dims=2, perplexity=30, verbose=TRUE, max_iter=500)
6 time.taken = system.time(Rtsne(X, dims=2, perplexity=30,
7                                verbose=TRUE, max_iter=500))
8 plot(tsne$Y, t='n', main="tsne")
9 text(tsne$Y, labels=train$label, col=colours[train$label])
```

# SNE and t-SNE (cont)

