Dimension Reduction Methods

Dr Sebnem Er

Statistical Sciences Department
University of Cape Town

August 29, 2023

Introduction

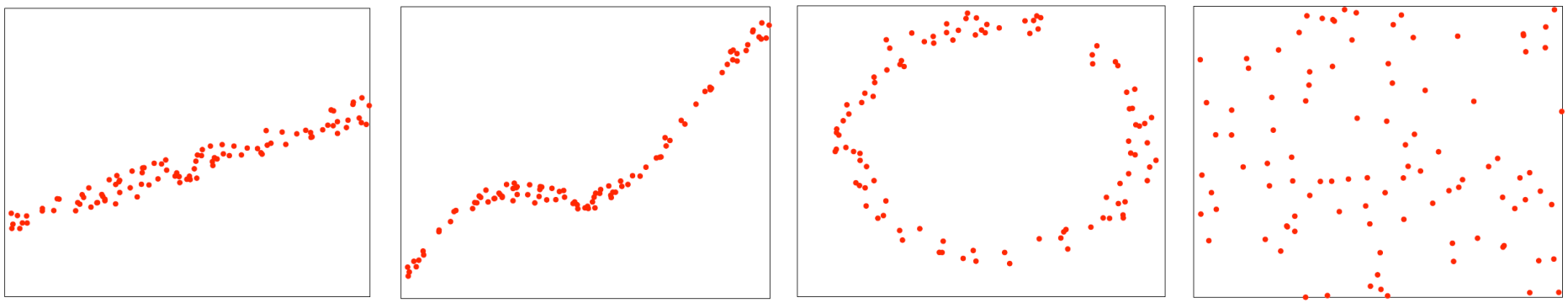
- Dimensionality reduction, enables data visualization and data pre-processing before supervised techniques are applied.
- Why reduce dimensions? Useful for:
  - Visualization
  - Further processing by machine learning algorithms
  - More efficient use of resources (e.g., time, memory, communication)
  - Statistical: fewer dimensions which implies better generalization
  - Statistical: in the case multicollinearity, if we do regression analysis, we end up with large variances for the parameter estimates, then we can use a subset of the principal components and the variance gets reduced.
  - Noise removal (improving data quality)

Dimension Reduction Methods

Linear and non-linear methods (manifold learning [2, 4]):

- PCA
- Kernel PCA
- MDS
- SOMs
- LLE (Local linear embedding)
- LE (Laplacian Eigenmap)
- HE (Hessian Eigenmap)
- ISOMAP
- t-SNE

Some examples



(a) (b) (c) (d)

Figure 1: (a) Linear (b) Nonlinear curve (c) Nonlinear (d) No obvious relationship

Ref: McGill School of Computer Science Machine Learning Lecture Notes

Principal Component Analysis

- Principal component analysis (PCA) is concerned with explaining the var-cov structure of a set of variables through a linear combinations of these variables.
- A PCA of a set of p variables (usually the original ones in standardized form) generates exactly p new set of uncorrelated variables, called *principal components* ($\mathbf{C}$), $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_p$, each of which is a linear combination of the variables in such a way that the first axis is in the direction containing most variation.

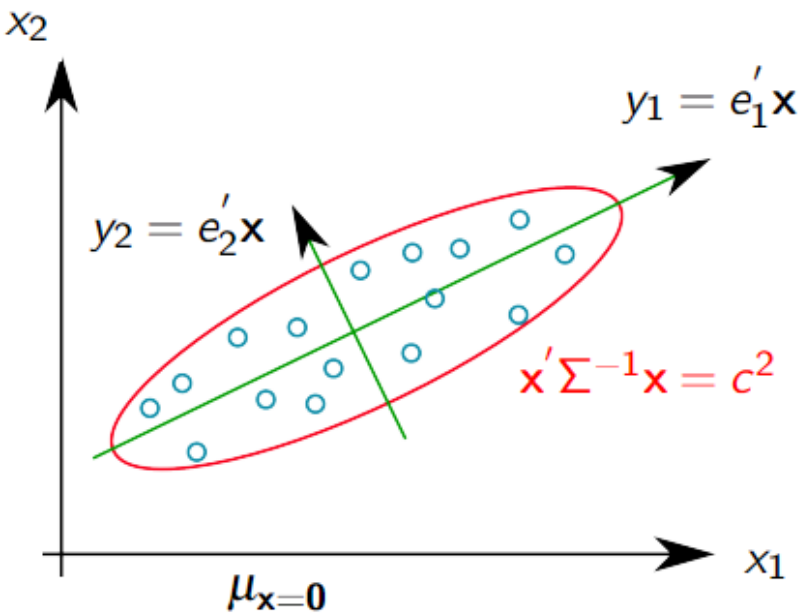
$$
\begin{array}{lcccl}
\mathbf{c}_1 & = & b_{11}\mathbf{x}_1 + b_{12}\mathbf{x}_2 + \ldots + b_{1p}\mathbf{x}_p & = & \mathbf{x}\mathbf{b}_1; \\
\mathbf{c}_2 & = & b_{21}\mathbf{x}_1 + b_{22}\mathbf{x}_2 + \ldots + b_{2p}\mathbf{x}_p & = & \mathbf{x}\mathbf{b}_2; \\
\vdots & & \vdots & & \vdots \\
\mathbf{c}_p & = & b_{p1}\mathbf{x}_1 + b_{p2}\mathbf{x}_2 + \ldots + b_{pp}\mathbf{x}_p & = & \mathbf{x}\mathbf{b}_p;
\end{array}
$$

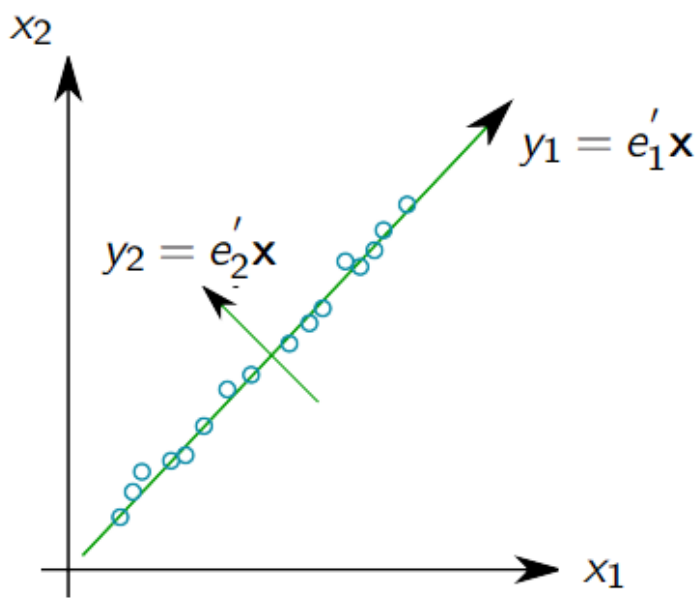
or in matrix form $\mathbf{C} = \mathbf{X}\mathbf{B}$

Principal Component Analysis

- The main aim of PCA is to find a low-dimensional subspace of the original dataset that contains as much information as possible. That is, if we were to project the data back from this transformation, the reconstructed data points lie as close as possible to the original values.
- Obviously, reducing the number of variables of a dataset comes with the expense of information loss, and less accuracy. How can we maintain as much information (total variation) as possible while reducing the number of variables?

Geometric View of PCA





Some matrix notation - data matrix

- Matrices in capital letter bold $\mathbf{X}$, vectors in small letter bold $\mathbf{x}_j$ where $j = 1, \ldots, p$, scalar in small letter no bold! X_{ij} where $i = 1, \ldots, n$
- $\mathbf{X}$ is a data matrix of order nxp (observation by variables) whose elements are raw scores of a subject i on predictor variable j.

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \ldots & X_{1p} \\ X_{21} & X_{22} & X_{12} & \ldots & X_{2p} \\ X_{31} & X_{32} & X_{33} & \ldots & X_{3p} \\ \vdots & & & & \\ X_{n1} & X_{n2} & X_{n3} & \ldots & X_{np} \end{bmatrix}$$

column vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \ldots & \mathbf{x}_p \end{bmatrix}$$

where $\mathbf{x}_j$ column vector gives the j-th variable's score for n observations.

Some matrix notation - var-cov matrix (population)

$$\Sigma_{XX} = \begin{bmatrix} \text{cov}(\mathbf{X}_1, \mathbf{X}_1) & \text{cov}(\mathbf{X}_1, \mathbf{X}_2) & \text{cov}(\mathbf{X}_1, \mathbf{X}_3) & \dots & \text{cov}(\mathbf{X}_1, \mathbf{X}_p) \\ \text{cov}(\mathbf{X}_2, \mathbf{X}_1) & \text{cov}(\mathbf{X}_2, \mathbf{X}_2) & \text{cov}(\mathbf{X}_3, \mathbf{X}_3) & \dots & \text{cov}(\mathbf{X}_2, \mathbf{X}_p) \\ \text{cov}(\mathbf{X}_3, \mathbf{X}_1) & \text{cov}(\mathbf{X}_3, \mathbf{X}_2) & \text{cov}(\mathbf{X}_3, \mathbf{X}_3) & \dots & \text{cov}(\mathbf{X}_3, \mathbf{X}_p) \\ \vdots & & & \dots & \vdots \\ \text{cov}(\mathbf{X}_p, \mathbf{X}_1) & \text{cov}(\mathbf{X}_p, \mathbf{X}_2) & \text{cov}(\mathbf{X}_p, \mathbf{X}_3) & \dots & \text{cov}(\mathbf{X}_p, \mathbf{X}_p) \end{bmatrix}$$

$$= \mathbf{X}_c' \mathbf{X}_c$$

Properties of var-cov matrix

- The matrix is symmetric since $\text{cov}(\mathbf{X}_j, \mathbf{X}_{j'}) = \text{cov}(\mathbf{X}_{j'}, \mathbf{X}_j)$
- Positive semi-definite
- Values of the matrix indicate how the variables vary together, $(-)$ values means they move in different directions, and $(+)$ values indicate they move in the same direction.

var-cov matrix (sample)

Often we do not know population var-cov values, we would then need to replace these with the sample var-cov values.

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}_c^{'}\mathbf{X}_c$$

where

$$\mathbf{X}_c = \mathbf{X} - \mathbf{1}_n\bar{\mathbf{x}}^{'} = \mathbf{C}$$

with

Some matrix notation - Example

$$\mathbf{X} = \begin{bmatrix} 102 & 4 \\ 104 & 5 \\ 101 & 7 \\ 93 & 1 \\ 100 & 3 \end{bmatrix} \qquad \mathbf{X}_c = \begin{bmatrix} 2 & 0 \\ 4 & 1 \\ 1 & 3 \\ -7 & -3 \\ 0 & -1 \end{bmatrix} \qquad \mathbf{S} = \frac{1}{5-1}\mathbf{X}_c'\mathbf{X}_c$$

$$\mathbf{S} = \begin{bmatrix} 17.5 & 7 \\ 7 & 5 \end{bmatrix}$$

Eigenvector - Eigenvalue pairs for the var-cov matrix

- Let $\mathbf{S}_x$ be a *pxp* square matrix associated with the p random variables $\mathbf{X}$.
- We have p number of principal components where $k = 1, \ldots, p$:

$$\mathbf{c}_k = b_{k1}\mathbf{x}_1 + b_{k2}\mathbf{x}_2 + \ldots + b_{kp}\mathbf{x}_p = \mathbf{x}\mathbf{b}_k$$

- In general, the coefficients of the principal components $\mathbf{c}_k$ are chosen so as to make its variance as large as possible, subject to the restrictions that it be uncorrelated with scores on $\mathbf{c}_1$ to $\mathbf{c}_{k-1}$.
- In order to eliminate the trivial solution, $b_{k1} = b_{k2} = \ldots = b_{kp} = \infty$, we require that the squares of the coefficients in any principal component sum to unity:

$$\sum_k b_{kj}^2 = \mathbf{b}_k^{'}\mathbf{b}_k = 1$$

Eigenvector - Eigenvalue pairs for the var-cov matrix

- variance as large as possible means we are maximizing the variance of the principal components given with (proof in [1]):

$$S_{c_k} = \mathbf{b}_k^{'} \mathbf{S}_x \mathbf{b}_k$$

- Using the Lagrange multipliers, we find that for $\mathbf{b}_1$:

$$L = \mathbf{b}_1^{'} \mathbf{S}_x \mathbf{b}_1 - \lambda(\mathbf{b}_1^{'} \mathbf{b}_1 - 1)$$

- Taking the derivative of L with respect to $\mathbf{b}_1$ gives us the following:

$$\frac{dL}{d\mathbf{b}_1} = 2\mathbf{S}_x \mathbf{b}_1 - 2\lambda \mathbf{b}_1 \quad \text{iff} \quad [\mathbf{S}_x - \lambda \mathbf{I}]\, \mathbf{b}_1 = \mathbf{0}$$

- Thus we have a set of p homogeneous equations in p unknowns. In order for this set of equations to have a nontrivial solution:

$$|\mathbf{S}_x - \lambda \mathbf{I}| = 0$$

Eigenvector - Eigenvalue pairs for the var-cov matrix

- The Lagrange multiplier for $\mathbf{b}_2$ has an additional term for the constraint of uncorrelated principal components:

$$L = \mathbf{b}_2^{'}\mathbf{S}_x\mathbf{b}_2 - \lambda(\mathbf{b}_2^{'}\mathbf{b}_2 - 1) - \theta(\mathbf{b}_2^{'}\mathbf{b}_1)$$

- Taking the derivative of L with respect to $\mathbf{b}_2$ gives us the following:

$$\frac{dL}{d\mathbf{b}_2} = 2\mathbf{S}_x\mathbf{b}_2 - 2\lambda\mathbf{b}_2 - \theta\mathbf{b}_1 \quad \text{iff} \quad 2\left[\mathbf{S}_x - \lambda\mathbf{I}\right]\mathbf{b}_2 = \theta\mathbf{b}_1$$

Since we know that $\theta = 0$ (a simple calculation of multiplying both sides with $\mathbf{b}_1^{'}$ will prove this [1, p.157]), we have $\left[\mathbf{S}_x - \lambda\mathbf{I}\right]\mathbf{b}_2 = \mathbf{0}$

- Thus we have a set of p homogeneous equations in p unknowns. In order for this set of equations to have a nontrivial solution:

$$|\mathbf{S}_x - \lambda\mathbf{I}| = 0$$

Eigenvector - Eigenvalue pairs for the var-cov matrix

- Computing the determinant with λ left in as an unknown produces pth degree polynomial in λ which, set equal to zero, constitutes the **characteristic equation**. Solving this equation for λ produces p roots, some of which may be zero if there is a **linear** dependence among the original variables [1, p.157].
- Then we substitute the nonzero λ eigenvalues ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$) in the matrix equation and solve the resulting set of equations for the coefficients $\mathbf{b}$ (eigenvector). Which sets of these p sets of coefficients should be for the first principal component ($\mathbf{c}_1$) [1, p.157]?
- The characteristic vector $\mathbf{b}$ are called the eigenvectors of matrix $\mathbf{S}_x$, and the elements of an eigenvector are the weights b_{kj} and are known as component loadings. The variance-covariance matrix of the principal components, are known as the eigenvalues of $\mathbf{S}_x$.

Eigenvector - Eigenvalue pairs for the var-cov matrix

- Remember we set out to maximize the variance of the first principal component, the coefficients of the first principal component will be the characteristic vector ($\mathbf{b}_1$) associated with the largest characteristic root (eigenvalue λ_1) [1, p.157].
- 2nd principal component is computed via the characteristic vector ($\mathbf{b}_2$) corresponding to the second largest characteristic root (eigenvalue λ_2), and so on [1, p.157].
- Eigenvalues, the variance explained by each principal component, are commonly plotted on a scree plot to show the decreasing rate at which variance is explained by each principal component.
- How do we choose how many components? (We will get to this.) First some manual calculations!

How do we obtain the characteristic roots and vectors of $\mathbf{S}_x$?

$$\mathbf{S}_x = \begin{bmatrix} 17.5 & 7 \\ 7 & 5 \end{bmatrix}$$

$$|\mathbf{S}_x - \lambda \mathbf{I}| = 0$$

$$det\left(\begin{bmatrix} 17.5 & 7 \\ 7 & 5 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = det\left(\begin{bmatrix} 17.5-\lambda & 7 \\ 7 & 5-\lambda \end{bmatrix} \right)$$

- Computing the determinant gives the following:

$$\begin{aligned} (17.5-\lambda)(5-\lambda) - 7*7 &= 0 \\ 87.5 - 17.5\lambda - 5\lambda + \lambda^2 - 49 &= 0 \\ \lambda^2 - 22.5\lambda + 38.5 = 0 & \end{aligned}$$

Anyone remembering how to do this?

$ax^2 + bx + c = 0 \implies x = \frac{-b \mp \sqrt{b^2-4ac}}{2a}$

How do we obtain the characteristic roots and vectors of $\mathbf{S}_x$?

- Where $\begin{array}{rcccl} \lambda_1 & = & \frac{22.5+\sqrt{22.5^2-4*1*38.5}}{2*1} & = & 20.63416 \\ \lambda_2 & = & \frac{22.5-\sqrt{22.5^2-4*1*38.5}}{2*1} & = & 1.865838 \end{array}$
- From $[\mathbf{S}_x - \lambda \mathbf{I}]\,\mathbf{b}_1 = \mathbf{0}$ and the constraint $\mathbf{b}_1^{'}\mathbf{b}_1 = 1$, we can obtain the corresponding eigenvector $\mathbf{b}_1$ for $\lambda_1 = 20.63416$ as:

$$\begin{bmatrix} 17.5 - 20.63416 & 7 \\ 7 & 5 - 20.63416 \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -3.13416 b_{11} + 7 b_{12} \\ 7 b_{11} - 15.63416 b_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} b_{11} & b_{12} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix} = b_{11}^2 + b_{12}^2 = 1$$

How do we obtain the characteristic roots and vectors of $\mathbf{S}_x$?

- Substitute $b_{11} = 2.333451 * b_{12}$ in $b_{11}^2 + b_{12}^2 = 1$ to obtain

$$\begin{aligned} 2.233452^2 * b_{12}^2 + b_{12}^2 &= 1 \\ 5.988307 b_{12}^2 &= 1 \\ b_{12} &= \sqrt{\frac{1}{5.988307}} \\ b_{12} &= 0.4086467 \\ b_{11} &= 2.233452 * 0.4086467 \\ b_{11} &= 0.9126928 \end{aligned}$$

- Then $\mathbf{b}_1$ and $\mathbf{b}_2$ (after substituting for $\lambda_2 = 1.865$) are as follows

$$\mathbf{b}_1 = \begin{bmatrix} 0.913 \\ 0.409 \end{bmatrix} \qquad \mathbf{b}_2 = \begin{bmatrix} -0.409 \\ 0.913 \end{bmatrix}$$

Principal Components

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{x}\mathbf{b}_1 = \quad 0.913\mathbf{x}_1 + 0.409\mathbf{x}_2; \\ \mathbf{c}_2 &= \mathbf{x}\mathbf{b}_2 = \quad -0.409\mathbf{x}_1 + 0.913\mathbf{x}_2; \end{aligned}$$

which can be obtained with:

Subject i	XC_1	XC_2	C_1	C_2
1	2	0	0.913*2+0.409*0 =1.826	-0.409*2 +0.913*0 =-0.818
2	4	1	4.061	-0.723
3	1	3	2.140	2.330
4	-7	-3	-7.618	0.124
5	0	-1	-0.409	-0.913

Table 1: Principal Component Scores

How to Perform PCA in R

1. Standardize the data matrix $\mathbf{X}$. First move the data to center of the coordinate system and then scale the data, usually to unit variance.
   - Principal components analysis requires multivariate normality. Transform your data if necessary where you have highly skewed variables for example. Outliers also should be dealt with.
2. Obtain the sample covariance matrix $\mathbf{S}_{xx}$.
   - If you have variables measured in different scales, you need to both center and scale the data so that a variable with larger values do not contribute more than the variables with smaller values. In that case perform the PCA on the correlation matrix instead of var-cov matrix.
3. Calculate the eigenvectors $\mathbf{e}$ that are the directions of the axes where there is the most variance and the corresponding eigenvalues λ of the covariance matrix where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$.
4. Rank the eigenvectors in order of their eigenvalues, largest to smallest, to obtain the principal components in order of significance.

How to Perform PCA in R

5. Since the main aim is to reduce the number of dimensions, decide which of these components to keep and which ones to discard. The eigenvectors of the components we keep are our feature vectors.

Why do we need data scaling?

- The distance formula is highly dependent on how features are measured.
- In particular, if certain features have a much larger range of values than the others, the distance measurements will be strongly dominated by the features with larger ranges.
- This is not a problem for datasets measured on the same scale.

min-max Normalization

This process transforms a feature such that all of its values fall in a range between 0 and 1. The formula for normalizing a feature is as follows:

$$X^{new} = \frac{X - min(X)}{max(X) - min(X)}$$

Normalized feature values can be interpreted as indicating how far, from 0 percent to 100 percent, the original value fell along the range between the original minimum and maximum.

z-score standardization

Another common transformation is called z-score standardization. The following formula subtracts the mean value of feature X, and divides the outcome by the standard deviation of X:

$$X^{new} = \frac{X - \bar{X}}{s_X}$$

This formula rescales each of the feature's values in terms of how many standard deviations they fall above or below the mean value. The resulting value is called a z-score. The z-scores fall in an unbound range of negative and positive numbers. Unlike the normalized values, they have no predefined minimum and maximum.

How many principal components?

1. Examine the increase in total variance explained with the inclusion of an additional principal component and discard all those that contribute very little to the total variance explained. In other words search for an elbow in the scree plot.
2. Determine an acceptable threshold of total percentage of variance explained and discard all those principal components that are below this threshold.
3. Check the eigenvalues and discard all those components that have eigenvalues less than 1 (when using standardized variables).
4. For all these examine the scree plot.

In R: Using prcomp() function

Refer to the RMD file.

```
> X1 = c(102, 104, 101, 93, 100)
> X2 = c(4, 5, 7, 1, 3)
> X1C = c(2, 4, 1, -7, 0)
> X2C = c(0, 1, 3, -3, -1)
> data = as.data.frame(X1,X2,X1C,X2C)
> prcomp(data[,3:4])
```

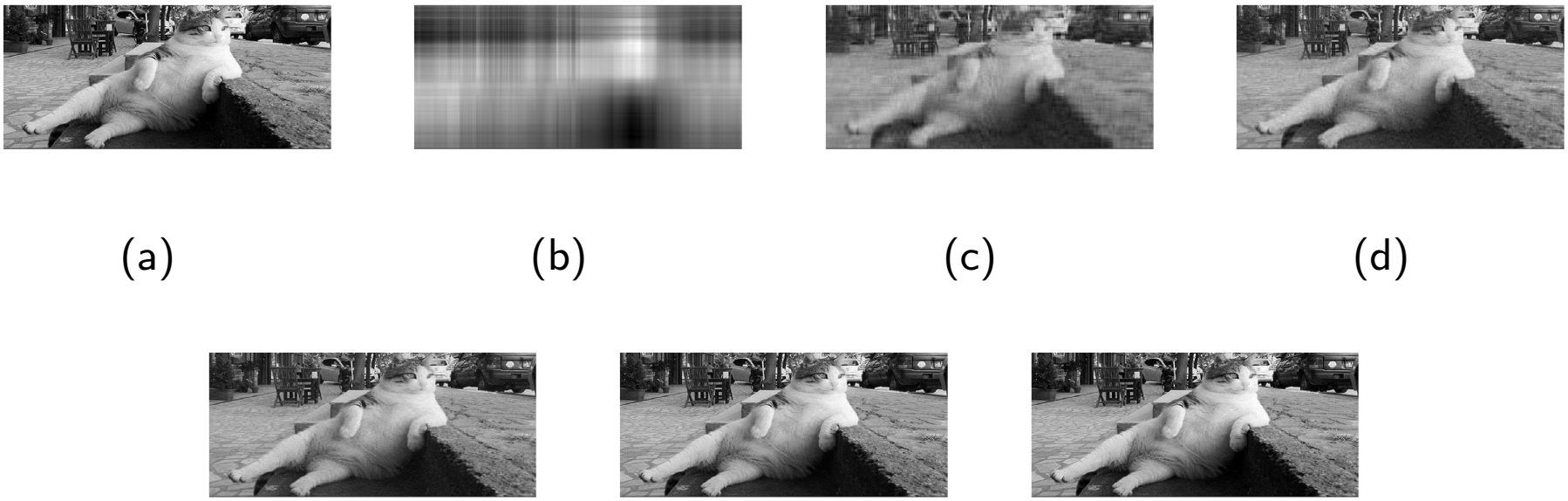
In R: using svd()

function Here the output produces 3 matrices, **d**, **U** and **V**. > `svd(cov(data[,1:2]))`

```
> svd(cov(data[,1:2]))
d
[1] 20.634162 1.865838
u
           [,1]       [,2]
[1,] -0.9126927 -0.4086467
[2,] -0.4086467 0.9126927
v
           [,1]       [,2]
[1,] -0.9126927 -0.4086467
[2,] -0.4086467 0.9126927
```

In R: Image compression

Here is Tombili the famous Turkish cat:



(a) (b) (c) (d)

(e) (f) (g)

Figure 2: (a) Original (b) 2 Principal Components (PCs) (c) 20 PCs (d) 50 PCs (e) 100 PCs (f) 200 PCs (e) 327 PCs

There is also Tommy the famous Turkish dog.

What is next?

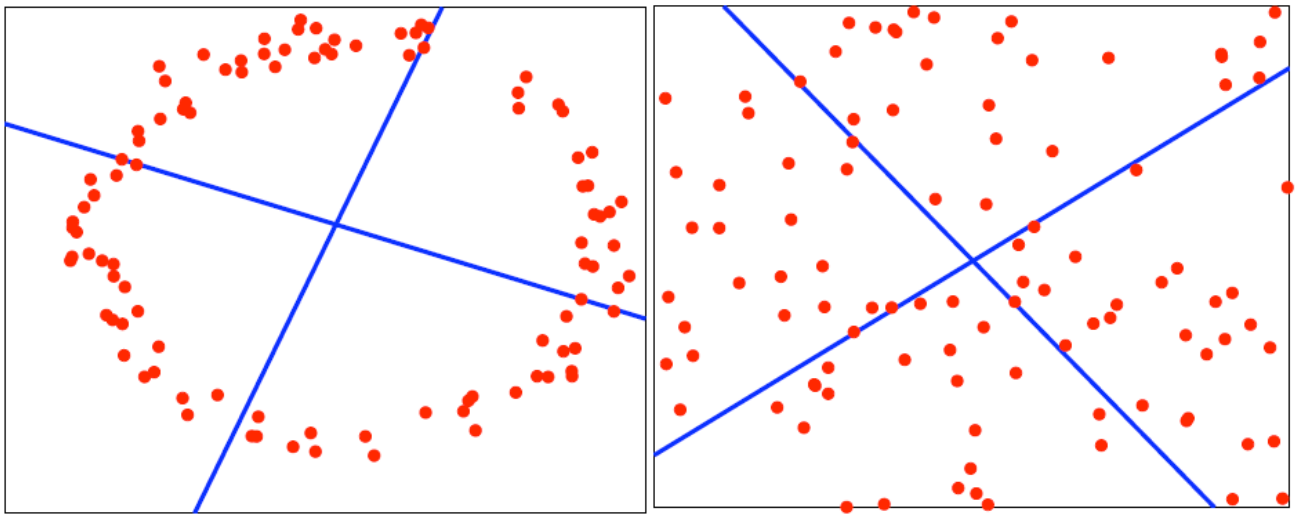


Figure 3: Difficult case

- How do we represent these kinds of datasets? Clearly the two linear PCs will not be the best. What can we do? [3]

References

[1] Richard J Harris. *A primer of multivariate statistics*. Psychology Press, 2001.

[2] Alan Julian Izenman. "Introduction to manifold learning". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.5 (2012), pp. 439–446.

[3] *TMcGill School of Computer Science Machine Learning Lecture Notes*. url https://www.cs.mcgill.ca/ dprecup/courses/ml.html, accessed 2023-08-27.

[4] Pavan Turaga, Rushil Anirudh, and Rama Chellappa. "Manifold Learning". In: *Computer Vision: A Reference Guide*. Ed. by Katsushi Ikeuchi. Cham: Springer International Publishing, 2021, pp. 784–789. ISBN: 978-3-030-63416-2. DOI: 10.1007/978-3-030-63416-2_824. URL: https://doi.org/10.1007/978-3-030-63416-2_824.