# Lecture 5: Dimensionality Reduction

Xenia Miscouridou

Imperial College London

Machine Learning Summer School
July 2022

# Outline

# Where does dimensionality reduction appear in machine learning?

Recall the setup so far

- Inputs: independent variables, predictors, covariates, $\boldsymbol{x}$
- Outputs: dependent variables, responses, labels, $\boldsymbol{y}$,
- Supervised learning: predict $\boldsymbol{y}$ from $\boldsymbol{x}$, examples are regression, classification.

$$\boldsymbol{x} \longrightarrow \boldsymbol{f} \longrightarrow \boldsymbol{y}$$

What if we do not have $\boldsymbol{y}$? What is the meaning of $\boldsymbol{f}$ in this case?

- Unsupervised learning: learn some useful information or patterns from $\boldsymbol{x}$, examples are dimensionality reduction and clustering.

# Why is it important in data science?

- What: Dimensionality Reduction is the process when we reduce the amount of random variables in a problem by keeping 'the important signal'.
- Why: Because of high dimensions!
- High Dimensions = A lot of features
- Examples

# Why is it important in data science?

- What: Dimensionality Reduction is the process when we reduce the amount of random variables in a problem by keeping 'the important signal'.
- Why: Because of high dimensions!
- High Dimensions = A lot of features
- Examples

# Why is it important in data science?

- What: Dimensionality Reduction is the process when we reduce the amount of random variables in a problem by keeping 'the important signal'.
- Why: Because of high dimensions!
- High Dimensions = A lot of features
- Examples

# Why is it important in data science?

- What: Dimensionality Reduction is the process when we reduce the amount of random variables in a problem by keeping 'the important signal'.
- Why: Because of high dimensions!
- High Dimensions = A lot of features
- Examples

# Where do we find high dimensional data?

Figure: Features per document are thousands of words


Figure: 13 million users in the UK and more than 5000 movies and tv shows

| | movie 1 | movie 2 | movie 3 | movie 4 | movie 5 | movie 6 |
|---|---|---|---|---|---|---|
| Tom | 5 | ? | ? | 1 | 3 | ? |
| George | ? | ? | 3 | 1 | 2 | 5 |
| Susan | 4 | 3 | 1 | ? | 5 | 1 |
| Beth | 4 | 3 | ? | 2 | 4 | 2 |

# More examples

Figure: Brain Imaging: many brain networks

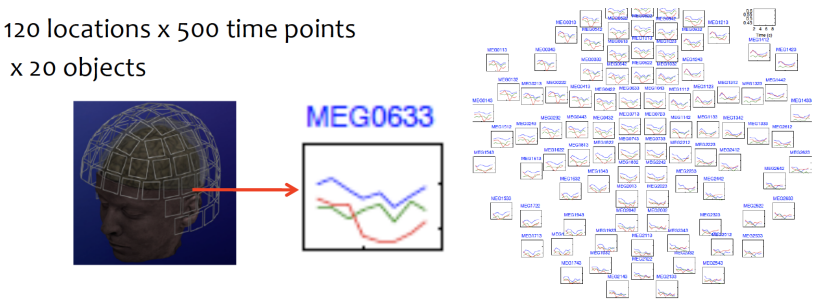
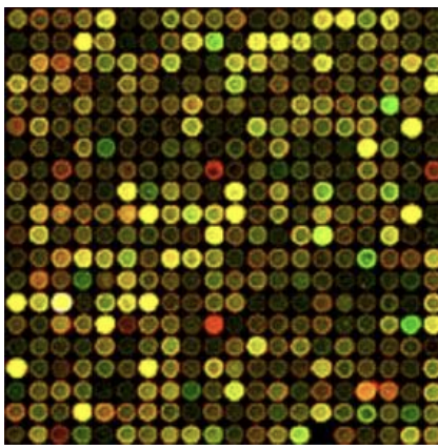



Figure: Human Genome: 10,000 genes x 1000 drugs x several species



# Outline

# Why is does dimensionality reduction very useful?

Problem It is hard to deal with many features that are possible noisy

Whereas learning lower dimensional representations of the data is useful and improves our ability to

- visualise the data
- remove dimensions that are not informative
- replace the original variables to more efficiently train a classifier or regression model
- reduce storage and computational costs
- interpret the data

# Why is does dimensionality reduction very useful?

Problem It is hard to deal with many features that are possible noisy

Whereas learning lower dimensional representations of the data is useful and improves our ability to

- visualise the data
- remove dimensions that are not informative
- replace the original variables to more efficiently train a classifier or regression model
- reduce storage and computational costs
- interpret the data

# Why is does dimensionality reduction very useful?

Problem It is hard to deal with many features that are possible noisy

Whereas learning lower dimensional representations of the data is useful and improves our ability to

- visualise the data
- remove dimensions that are not informative
- replace the original variables to more efficiently train a classifier or regression model
- reduce storage and computational costs
- interpret the data

# Why is does dimensionality reduction very useful?

**Problem** It is hard to deal with many features that are possible noisy

Whereas learning lower dimensional representations of the data is useful and improves our ability to

- visualise the data
- remove dimensions that are not informative
- replace the original variables to more efficiently train a classifier or regression model
- reduce storage and computational costs
- interpret the data

# Why is does dimensionality reduction very useful?

**Problem** It is hard to deal with many features that are possible noisy

Whereas learning lower dimensional representations of the data is useful and improves our ability to

- visualise the data
- remove dimensions that are not informative
- replace the original variables to more efficiently train a classifier or regression model
- reduce storage and computational costs
- interpret the data

# Why is does dimensionality reduction very useful?

**Problem** It is hard to deal with many features that are possible noisy

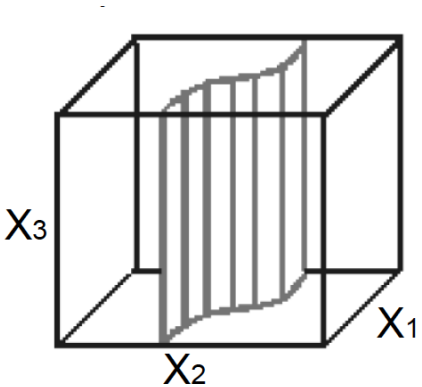
Whereas learning lower dimensional representations of the data is useful and improves our ability to

- visualise the data
- remove dimensions that are not informative
- replace the original variables to more efficiently train a classifier or regression model
- reduce storage and computational costs
- interpret the data

# How do I do dimensionality reduction?

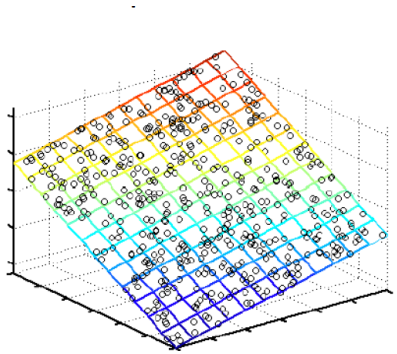
- Methods to transform the high-dimensional data to a space of less dimensions
  - Feature Selection: Remove features that are irrelevant (left)
  - Latent Feature Extraction: Some combination of features provides a more efficient representation than observed features (right)

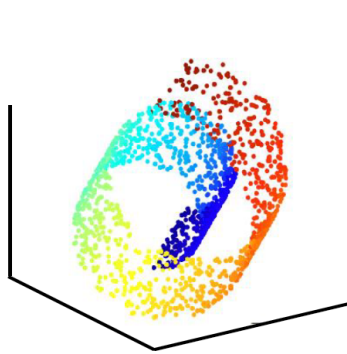
Figure: Feature Selection




$X_3$ - Irrelevant

Figure: Feature Extraction





# How do I do latent feature extraction?

- Combinations of observed features provide more efficient representations, and capture underlying relations in data
- Example topics (sports, science, news, etc.) instead of documents
- Often may not have physical meaning
- The transformation may be linear
  - Principal Components Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis
- Or nonlinear
  - Embedding methods
  - Mappings using ISOMAP, eigenmaps etc

# How do I do latent feature extraction?

- Combinations of observed features provide more efficient representations, and capture underlying relations in data
- Example topics (sports, science, news, etc.) instead of documents
- Often may not have physical meaning
- The transformation may be linear
  - Principal Components Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis
- Or nonlinear
  - Embedding methods
  - Mappings using ISOMAP, eigenmaps etc

# How do I do latent feature extraction?

- Combinations of observed features provide more efficient representations, and capture underlying relations in data
- Example topics (sports, science, news, etc.) instead of documents
- Often may not have physical meaning
- The transformation may be linear
  - Principal Components Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis
- Or nonlinear
  - Embedding methods
  - Mappings using ISOMAP, eigenmaps etc

# How do I do latent feature extraction?

- Combinations of observed features provide more efficient representations, and capture underlying relations in data
- Example topics (sports, science, news, etc.) instead of documents
- Often may not have physical meaning
- The transformation may be linear
  - Principal Components Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis
- Or nonlinear
  - Embedding methods
  - Mappings using ISOMAP, eigenmaps etc

# How do I do latent feature extraction?

- Combinations of observed features provide more efficient representations, and capture underlying relations in data
- Example topics (sports, science, news, etc.) instead of documents
- Often may not have physical meaning
- The transformation may be linear
  - Principal Components Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis
- Or nonlinear
  - Embedding methods
  - Mappings using ISOMAP, eigenmaps etc

# Outline

# Principal Component Analysis

Figure: Intrinsically lower dimensional than the dimension of the space

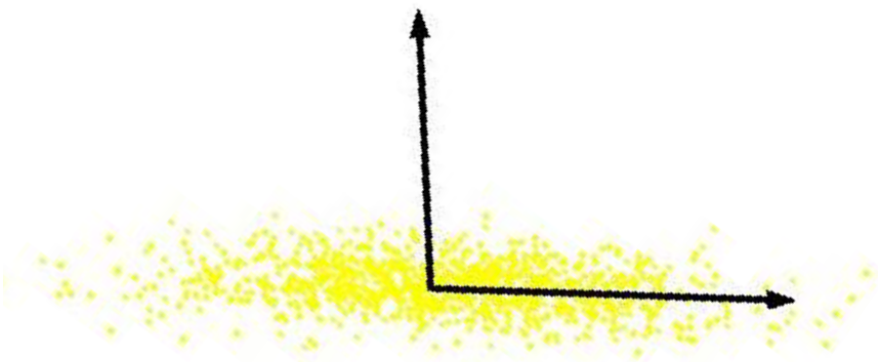
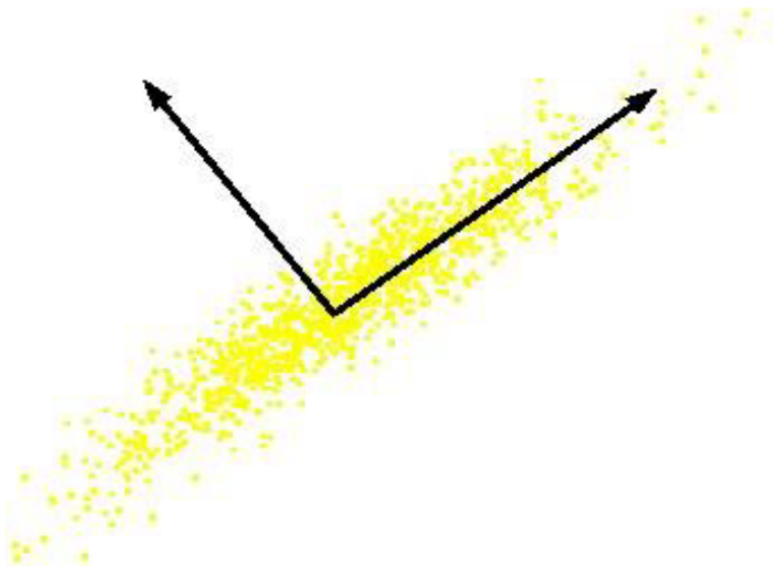
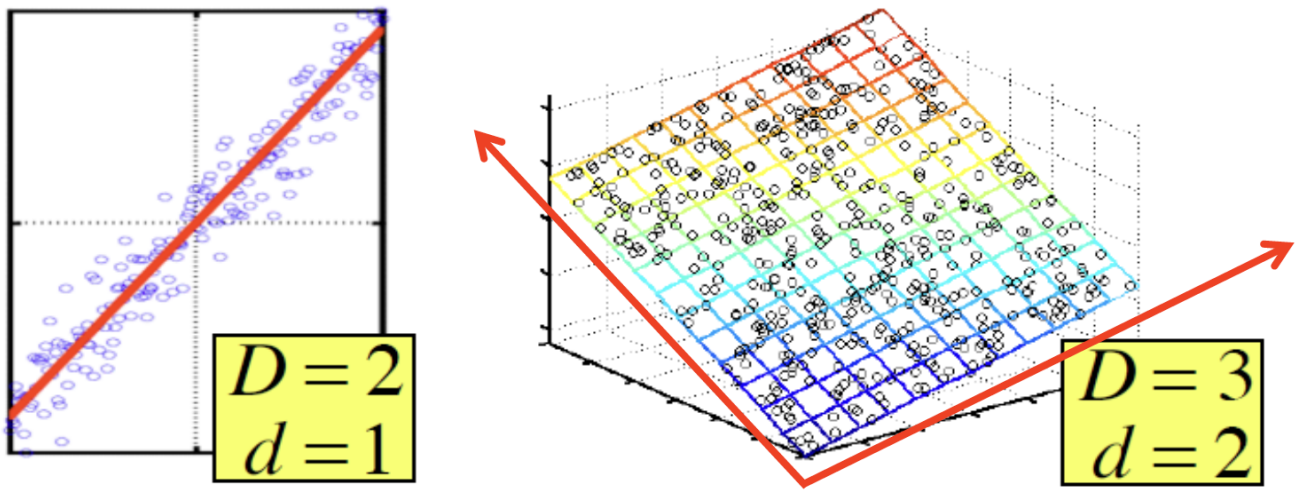


Figure: If we rotate data, again only one coordinate is more important



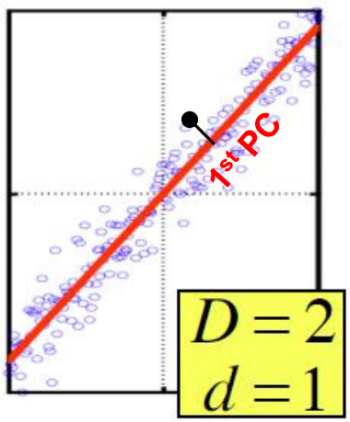
- Can we transform the features so that we only need to preserve one latent feature? Find the linear projection so that projected data is uncorrelated

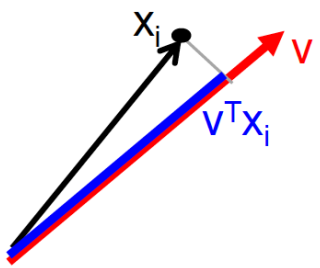
# Principal Component Analysis




- Assumptions: Data lies on or near a low d-dimensional linear subspace
- Identifying the axes is known as Principal Components Analysis, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition)

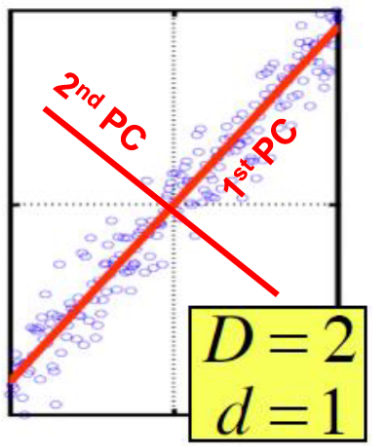
# Principal Components - 1st

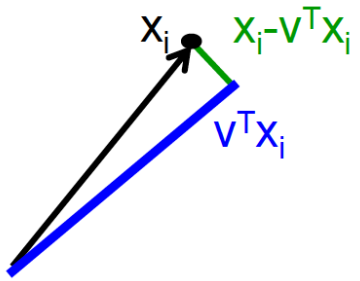



- Principal Components (PC) are orthogonal directions that capture most of the variance in the data
- 1st PC direction of greatest variability in data
- Projection of data points along 1st PC discriminate the data most along any one direction
- Take a data point $x_i$ (D-dimensional vector)
- Projection of $x_i$ onto the 1st PC $v$ is $v^T x_i$

# Principal Components - 2nd



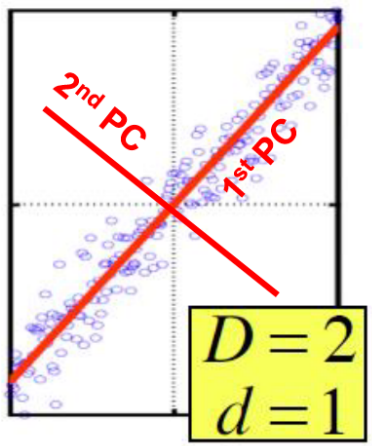



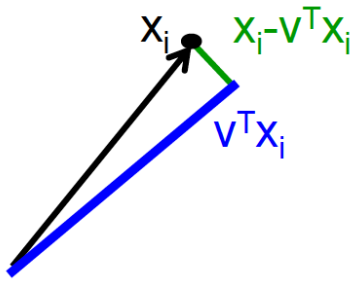

- Principal Components (PC) are orthogonal directions that capture most of the variance in the data
- 1st PC direction of greatest variability in data
- 2nd PC Next orthogonal (uncorrelated) direction of greatest variability
- (remove all variability in first direction, then find next direction of greatest variability)
- continue to find up to the $d^{th}$ PC

Finally

- instead of original $D$-dim data features $x_i = [x_i^1, x_i^2, \ldots, x_i^D]$
- use $d$-dim projections $\tilde{x}_i = [v_1^T x_i, \ldots, v_d^T x_i]$

# Principal Components - 2nd



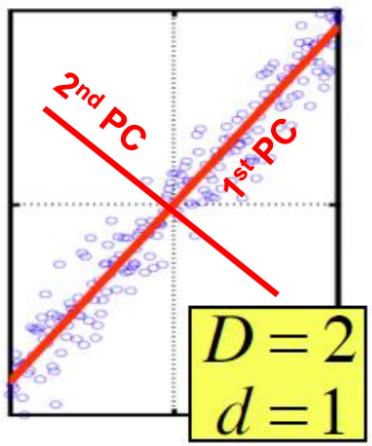



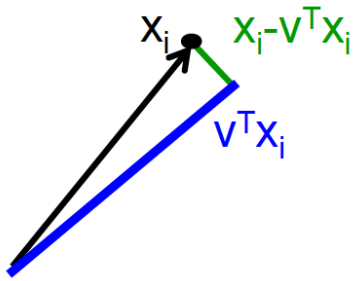

- Principal Components (PC) are orthogonal directions that capture most of the variance in the data
- 1st PC direction of greatest variability in data
- 2nd PC Next orthogonal (uncorrelated) direction of greatest variability
- (remove all variability in first direction, then find next direction of greatest variability)
- continue to find up to the $d^{th}$ PC

Finally

- instead of original $D$-dim data features $x_i = [x_i^1, x_i^2, \ldots, x_i^D]$
- use $d$-dim projections $\tilde{x}_i = [v_1^T x_i, \ldots, v_d^T x_i]$

# Principal Components - 2nd



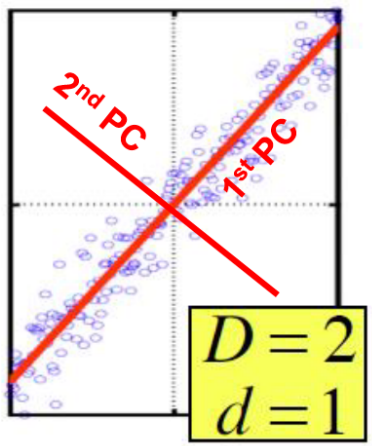



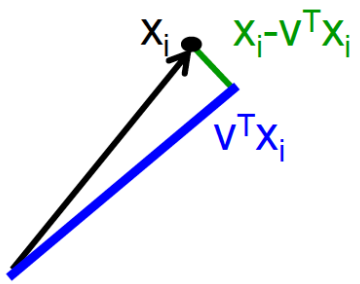

- Principal Components (PC) are orthogonal directions that capture most of the variance in the data
- 1st PC direction of greatest variability in data
- 2nd PC Next orthogonal (uncorrelated) direction of greatest variability
- (remove all variability in first direction, then find next direction of greatest variability)
- continue to find up to the $d^{th}$ PC

Finally

- instead of original $D$-dim data features $x_i = [x_i^1, x_i^2, \ldots, x_i^D]$
- use $d$-dim projections $\tilde{x}_i = [v_1^T x_i, \ldots, v_d^T x_i]$

# Principal Components - 2nd



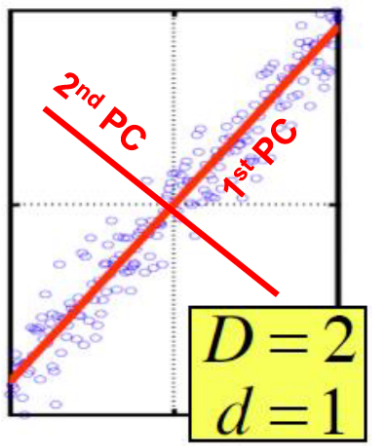



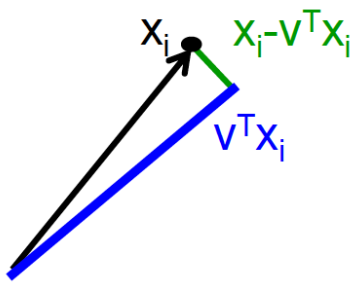

- Principal Components (PC) are orthogonal directions that capture most of the variance in the data
- 1st PC direction of greatest variability in data
- 2nd PC Next orthogonal (uncorrelated) direction of greatest variability
- (remove all variability in first direction, then find next direction of greatest variability)
- continue to find up to the $d^{th}$ PC

Finally

- instead of original $D$-dim data features $x_i = [x_i^1, x_i^2, \ldots, x_i^D]$
- use $d$-dim projections $\tilde{x}_i = [v_1^T x_i, \ldots, v_d^T x_i]$

# Principal Components - 2nd




- Principal Components (PC) are orthogonal directions that capture most of the variance in the data
- 1st PC direction of greatest variability in data
- 2nd PC Next orthogonal (uncorrelated) direction of greatest variability
- (remove all variability in first direction, then find next direction of greatest variability)
- continue to find up to the $d^{th}$ PC

Finally

- instead of original $D$-dim data features $x_i = [x_i^1, x_i^2, \ldots, x_i^D]$
- use $d$-dim projections $\tilde{x}_i = [v_1^T x_i, \ldots, v_d^T x_i]$

# How do I find the vectors v?

- Let $v_1, \ldots, v_d$ denote the principal components, which are orthogonal and have unit norm

$$v_i^T v_j = 0, i \neq j,$$
$$v_i^T v_i = 1 \quad (1)$$

- Assuming data are centered $X = [x_1 x_2 \ldots]$ find vector that maximizes sample variance of projections given by

$$\frac{1}{n}\sum_{i=1}^{n}(v^T x_i)^2 = v^T X X^T v$$

- Solve

$$\max v^T X X^T v \text{ s.t. } v^T v = 1 \iff \text{Lagrangian: } \max_v \; v^T X X^T v - \lambda v^T v$$

- Setting $\partial/\partial v = 0$ and $\left(XX^T - \lambda I\right) v = 0$

This wraps the constraints into the objective function $(XX^T)v = \lambda v$

# How do I find the vectors v?

$$(XX^T)v = \lambda v$$

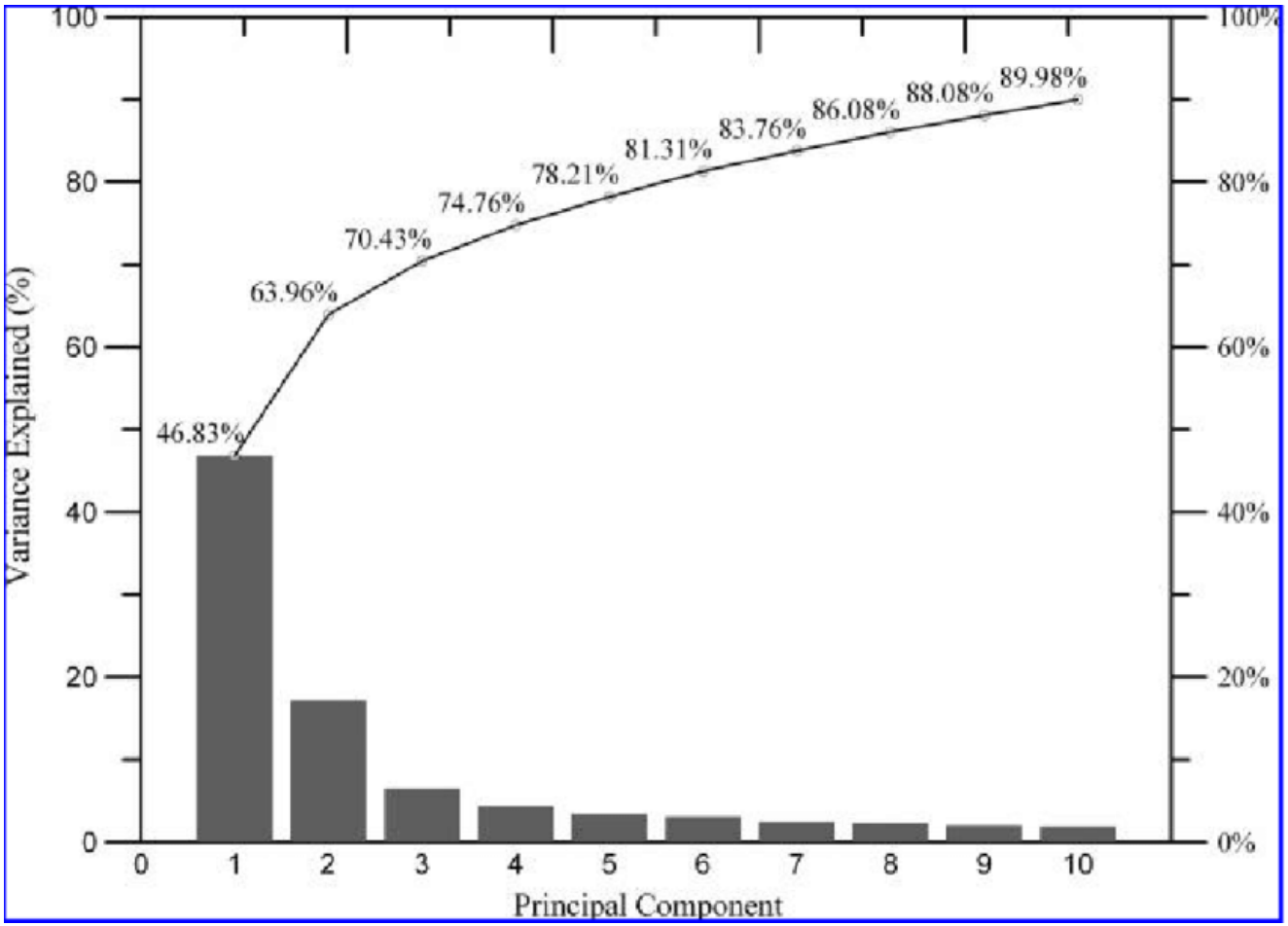
- The above can be recognised as the eigenvalue formula, i.e. v is the eigenvector of sample correlation/covariance matrix $XX^T$
- Then, the sample variance of projection$= v^T XX^T v = \lambda v^T v = \lambda$
- Thus, the values $\lambda$ that are called eigenvalues, denote the amount of variability captured along that dimension
- $\lambda_1 > \lambda_2 > \lambda_3 > \dots$
- The first Principal Component $v_1$ is the eigenvector associated with the largest eigenvalue $\lambda_1$ and hence the largest variability
- The second Principal Component $v_2$ is the eigenvector associated with the second largest eigenvalue $\lambda_2$

# How do I find the vectors v?

- The equation we end up is $(XX^T)v = \lambda v \iff (XX^T - \lambda\mathbb{I})v = 0$
- A non zero solution is possible only if $det(XX^T - \lambda\mathbb{I}) = 0$ Characteristic Equation
- This is a $D$th order equation in $\lambda$ and can have at most $D$ solutions (that's why we can only have at most $D$ PCs)
- Once the values $\lambda$ (eigenvalues) are computed then we need to solve for $v$ our Principal Components (eigenvectors) using $(XX^T - \lambda\mathbb{I})v = 0$
- So overall we have an Exact Solution

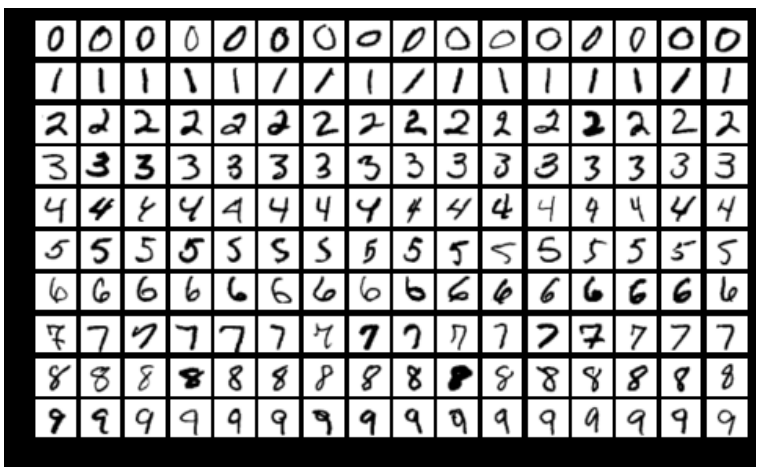
# Dimensionality Reduction using PCA

The more components you include the more variance you explain and the less information you loose but there's a trade off



# Outline

# PCA on MNIST



# PCA on real genomes: European genetic variations

Population structure within Europe as this is illustrated by the first and second principal components of the observed genetic variation. The resulting figure looks very similar to the geographic map of Europe.

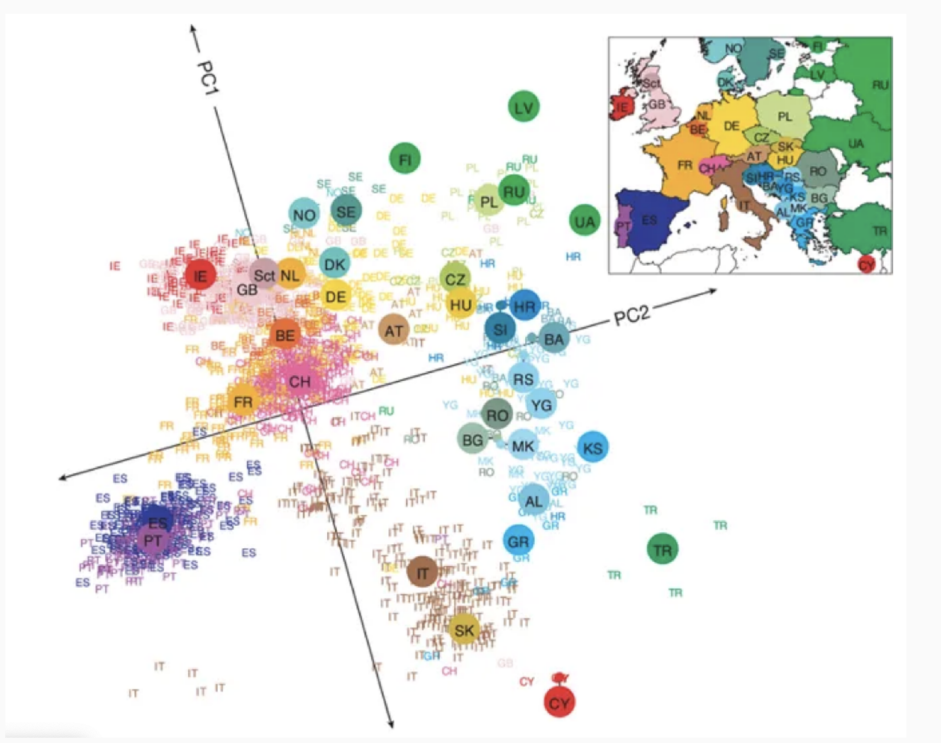



Figure: From "Genes mirror geography within Europe" published in Nature 2008
https://www.nature.com/articles/nature07331

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# PCA Summary

- PCA is a very common technique for linear dimensionality reduction
- The first few principal components often provide a good reconstruction of the data in small dimensions and explain a large proportion of the overall variance
- Choosing the dimensions of the lower dimensional space depends on the overall objective of the task in hand
- It is often used as a feature extraction for further analysis
- It assumes that the observations are real-valued with no missing observations
- Several extensions exist to deal with other kinds of data (e.g. probabilistic PCA) and for non-linear dimensionality reduction (e.g. kernel PCA)

# Other References

Other types of dimensionality reduction are

- Non-negative matrix factorisation (NMF): linear method
- Linear discriminant analysis (LDA) : linear method used when we have labelled data
- t-distributed Stochastic Neighbour Embedding (t-SNE): non-linear method mostly used for data visualisation and widely used in image processing and NLP
- Umap: similar to t-sne but also for more general non-linear reduction

  https://umap-learn.readthedocs.io/en/latest/

  https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

# Other References

Other types of dimensionality reduction are

- Non-negative matrix factorisation (NMF): linear method
- Linear discriminant analysis (LDA) : linear method used when we have labelled data
- t-distributed Stochastic Neighbour Embedding (t-SNE): non-linear method mostly used for data visualisation and widely used in image processing and NLP
- Umap: similar to t-sne but also for more general non-linear reduction

  https://umap-learn.readthedocs.io/en/latest/

  https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

# Other References

Other types of dimensionality reduction are

- Non-negative matrix factorisation (NMF): linear method
- Linear discriminant analysis (LDA) : linear method used when we have labelled data
- t-distributed Stochastic Neighbour Embedding (t-SNE): non-linear method mostly used for data visualisation and widely used in image processing and NLP
- Umap: similar to t-sne but also for more general non-linear reduction

  https://umap-learn.readthedocs.io/en/latest/

  https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

# Other References

Other types of dimensionality reduction are

- Non-negative matrix factorisation (NMF): linear method
- Linear discriminant analysis (LDA) : linear method used when we have labelled data
- t-distributed Stochastic Neighbour Embedding (t-SNE): non-linear method mostly used for data visualisation and widely used in image processing and NLP
- Umap: similar to t-sne but also for more general non-linear reduction

  https://umap-learn.readthedocs.io/en/latest/

  https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

# Other References

Other types of dimensionality reduction are

- Non-negative matrix factorisation (NMF): linear method
- Linear discriminant analysis (LDA) : linear method used when we have labelled data
- t-distributed Stochastic Neighbour Embedding (t-SNE): non-linear method mostly used for data visualisation and widely used in image processing and NLP
- Umap: similar to t-sne but also for more general non-linear reduction

  https://umap-learn.readthedocs.io/en/latest/

  https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

# t-sne on MNIST