Dimensionality Reduction Techniques

By

Prof. (Dr.) Premanand P. Ghadekar

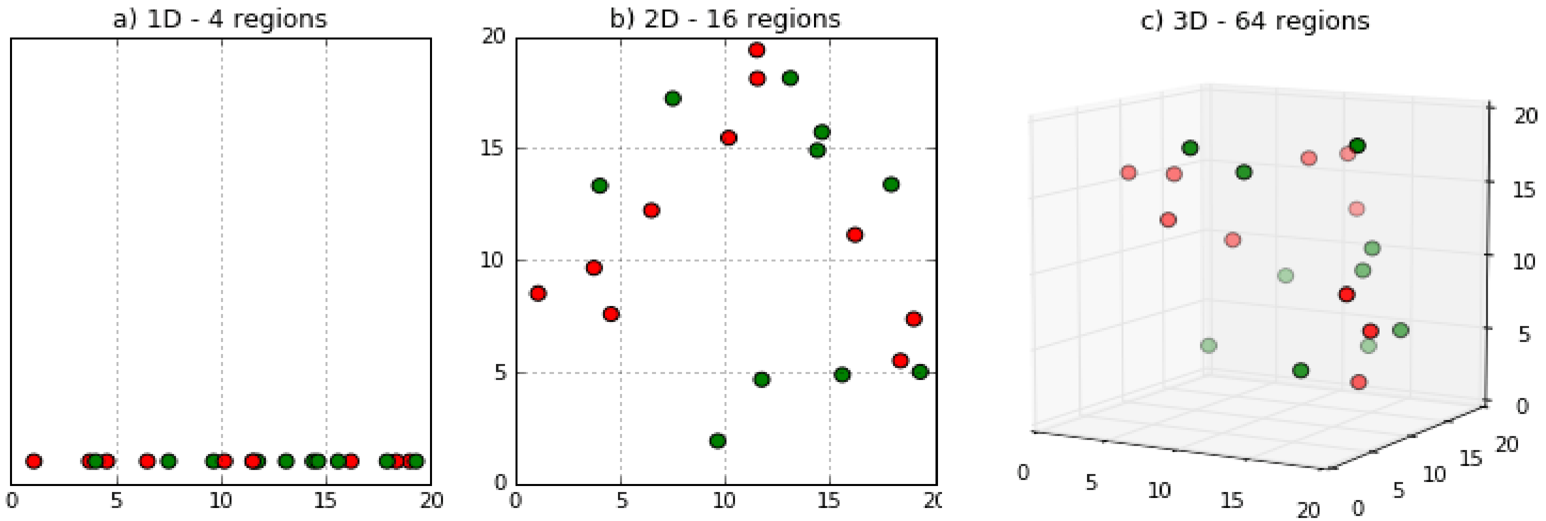
Outline

- Introduction
- Dimensionality Reduction
- Feature Selection
- Feature Extraction Techniques
  - Principal Component Analysis-PCA
  - Kernel Principal Component Analysis-KPCA
  - Independent Component Analysis
  - Singular Value Decomposition-SVD
  - Applications

Introduction

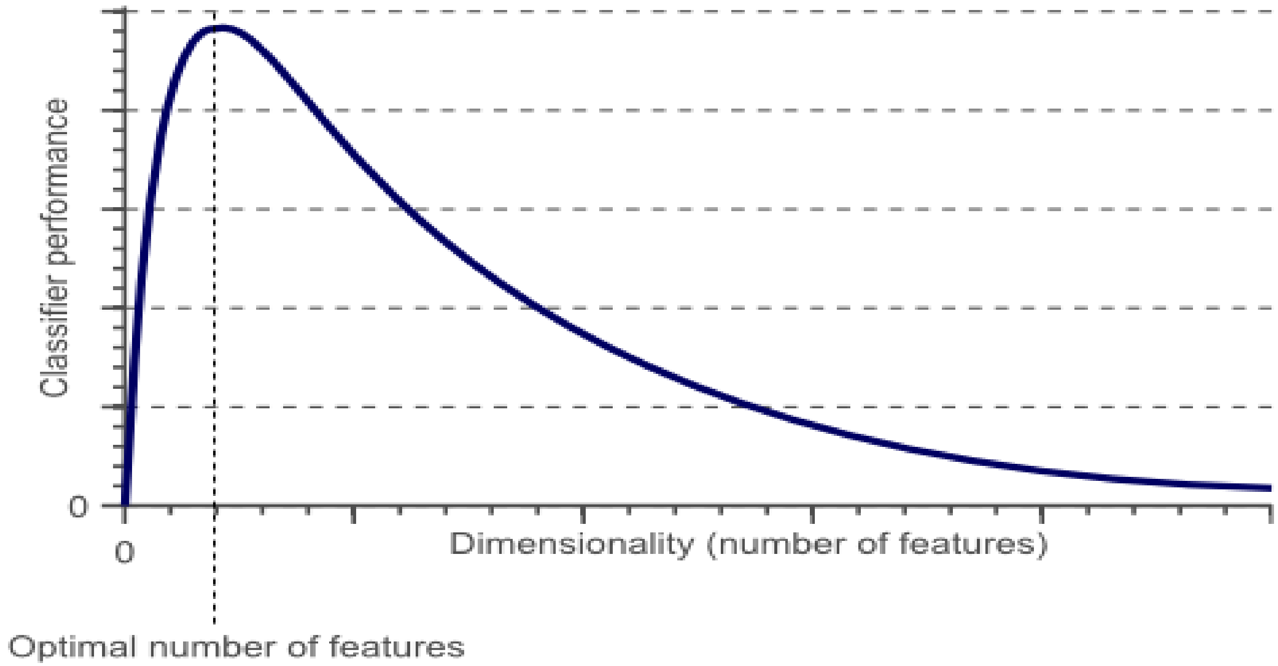
- The number of **input variables** or **features** for a dataset is referred to as its **dimensionality**.
- Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset.
- More input features often make a predictive modeling task more challenging to model.
- High-dimensionality statistics and dimensionality reduction techniques are often used for **data visualization.**
- Used **to reduce computational complexity and to simplify a classification or regression dataset** in order to better fit a predictive model.

Curse of Dimensionality



- In the above example, **data points lying in one dimension** need only 4 spaces for describing any of the points.
- In the second image, with **an increase in dimension by only one** (2-dimensional), the number of spaces increases to 16. And in the third image, with another addition of dimension, the number of spaces rises to 64.
- This shows that as **the number of dimensions increases, the amount of data needed to generalize increases exponentially.**

Curse of Dimensionality



In the image shown above, it can be seen that **with an increase in dimensions beyond the optimum number, classifier performance goes on decreasing. (Overfitting Problem).**

Problem With Many Input Variables

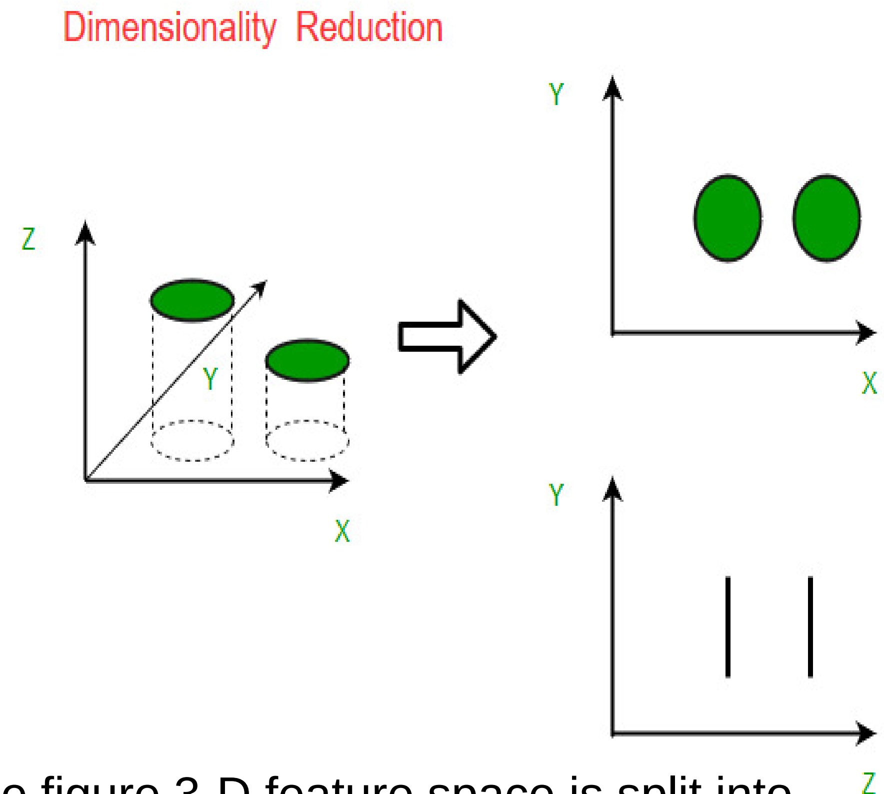
- The performance of machine learning algorithms can degrade with too many input variables.
- Having a large number of dimensions in the feature space can mean that the volume of that space is very large.
- Huge amount of data causes **the overfitting problem.**
- Therefore, it is often desirable to reduce the number of input features.
- It is a data preparation technique performed on data prior to modeling.
- It might be performed after data cleaning and data scaling and before training a predictive model.

Why Dimensionality Reduction?

- By reducing the dimensions of the features, **the space required to store the dataset also gets reduced.**
- **Less Computation training time** is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in **visualizing the data quickly.**
- It **removes the redundant features** (if present) by taking care of multicollinearity.
- **Multicollinearity** is the occurrence of high intercorrelations among two or more independent variables in a multiple regression model.

Techniques for Dimensionality Reduction

- Feature Selection Methods
- Feature Extraction Methods
- Autoencoder Methods
- Independent Component Analysis
- Principal Component Analysis
- Singular Value Decomposition etc.



In the figure 3-D feature space is split into two 1-D feature spaces

Feature Selection Methods

- It uses scoring or statistical methods to select which features to keep and which features to delete.
- It is used to remove **"irrelevant"** features that do not help much with the classification problem.
- **There are three general classes of feature selection algorithms:**
  - **Filter**
  - **Wrapping**
  - **Embedding**

Feature Selection Methods-Filter Methods

- Filter feature selection methods **apply a statistical measure** to assign a scoring to each feature.
- The **features are ranked by the score** and either selected to be kept or removed from the dataset.
- The methods are often univariate and consider the feature independently, or with regard to the dependent variable.
- Some examples of some filter methods include **the Chi squared test, information gain** and **correlation coefficient scores.**

Feature Selection Methods-Filter Methods

What is Correlation?

Variables within a dataset can be related for lots of reasons.

- For example:
  - One variable could cause or depend on the values of another variable.
  - One variable could be lightly associated with another variable.
  - Two variables could depend on a third unknown variable.
- A correlation could be positive, Neutral or Negative
  - **Positive Correlation:** both variables change in the same direction.
  - **Neutral Correlation:** No relationship in the change of the variables.
  - **Negative Correlation:** variables change in opposite directions.

Depending what is known about the relationship and the distribution of the variables, **different correlation scores can be calculated.**

Backward Feature Elimination

- The **backward feature elimination technique** is mainly used while developing **Linear Regression** or **Logistic Regression model.**
- **Below steps are performed in this technique:**
- In this technique, firstly, all the n variables of the given dataset are taken to train the model.
- The performance of the model is checked.
- Now we will remove one feature each time and train the model on n-1 features for n times, and will compute the performance of the model.
- We will check the variable that has made the smallest or no change in the performance of the model, and then we will drop that variable or features; after that, we will be left with n-1 features.
- Repeat the complete process until no feature can be dropped.

Forward Feature Selection

- Forward feature selection follows the inverse process of the backward elimination process.
- It means, in this technique, we don't eliminate the feature; instead, we will find the best features that can produce the highest increase in the performance of the model.
- **Below steps are performed in this technique:**
- We start with a single feature only, and progressively we will add each feature at a time.
- Here we will train the model on each feature separately.
- The feature with the best performance is selected.
- The process will be repeated until we get a significant increase in the performance of the model.

Missing Value Ratio

- If a dataset has too many missing values, then we drop those variables as they do not carry much useful information.
- To perform this, we can set a **threshold level**, and if a variable has missing values more than that threshold, we will drop that variable.
- The higher the threshold value, the more efficient the reduction.

Low Variance Filter

- As same as missing value ratio technique, data columns with some changes in the data have less information.
- Therefore, we need to calculate the variance of each variable, and all data columns with variance lower than a given threshold are dropped because low variance features will not affect the target variable.

High Correlation Filter

- High Correlation refers to the case when two variables carry approximately similar information.
- Due to this factor, the performance of the model can be degraded.
- This correlation between the independent numerical variable gives the calculated value of the correlation coefficient.
- If this value is higher than the threshold value, we can remove one of the variables from the dataset.
- We can consider those variables or features that show a high correlation with the target variable.

Random Forest

- Random Forest is a popular and very useful feature selection algorithm in machine learning.
- This algorithm contains an in-built feature importance package, so we do not need to program it separately.
- In this technique, we need to generate a large set of trees against the target variable, and with the help of usage statistics of each attribute, we need to find the subset of features.
- Random forest algorithm takes only numerical variables, so we need to convert the input data into numeric data using hot encoding.

Factor Analysis

- **Factor analysis** is a technique in which each variable is kept within a group according to the correlation with other variables, it means **variables within a group can have a high correlation between themselves, but they have a low correlation with variables of other groups.**
- We can understand it by an example, such as if **we have two variables Income and spend.**
- These two variables have a high correlation, which means people with high income spends more, and vice versa.
- So, such variables are put into a group, and that group is **known as the factor.**
- The number of these factors will be reduced as compared to the original dimension of the dataset.

Auto-encoders

- One of the popular methods of dimensionality reduction is auto-encoder, which is **a type of ANN or artificial neural network** , and its main aim is to copy the inputs to their outputs.
- In this, **the input is compressed into latent-space representation,** and output is occurred using this representation.
- It has mainly two parts:
- **Encoder:** The function of the encoder is to compress the input to form the latent-space representation.
- **Decoder:** The function of the decoder is to recreate the output from the latent-space representation.
- **A latent space,** is an embedding of a set of items within a manifold in which items which resemble each other more closely are positioned closer to one another in the latent space.

Feature Extraction

- Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions.
- This approach is useful when we want to keep the whole information but use fewer resources while processing the information.
- **Some common feature extraction techniques are:**
  - Principal Component Analysis-PCA
  - Kernel Principal Component Analysis-KPCA
  - Independent Component Analysis
  - Singular Value Decomposition-SVD

Principal Component Analysis

What is Principal Component Analysis?

- Analysis of n-dimensional data
- Observes correspondence between different dimensions
- Determines principal dimensions along which the **variance** of the data is high

Why Principal Component Analysis?

- Determines a (lower dimensional) basis to represent the data
- Useful compression mechanism
- Useful for decreasing dimensionality of high dimensional data

Principal Component Analysis

- Principal Component Analysis is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features.
- These new transformed features are called the Principal Components.
- It is one of the popular tools that is used for exploratory data analysis and predictive modeling.
- PCA works by considering the variance of each attribute.
- High Variance shows the good split between the classes, and hence it reduces the dimensionality.
- Some real-world applications of PCA are *image processing, movie recommendation system, optimizing the power allocation in various communication channels.*

Principal Component Analysis

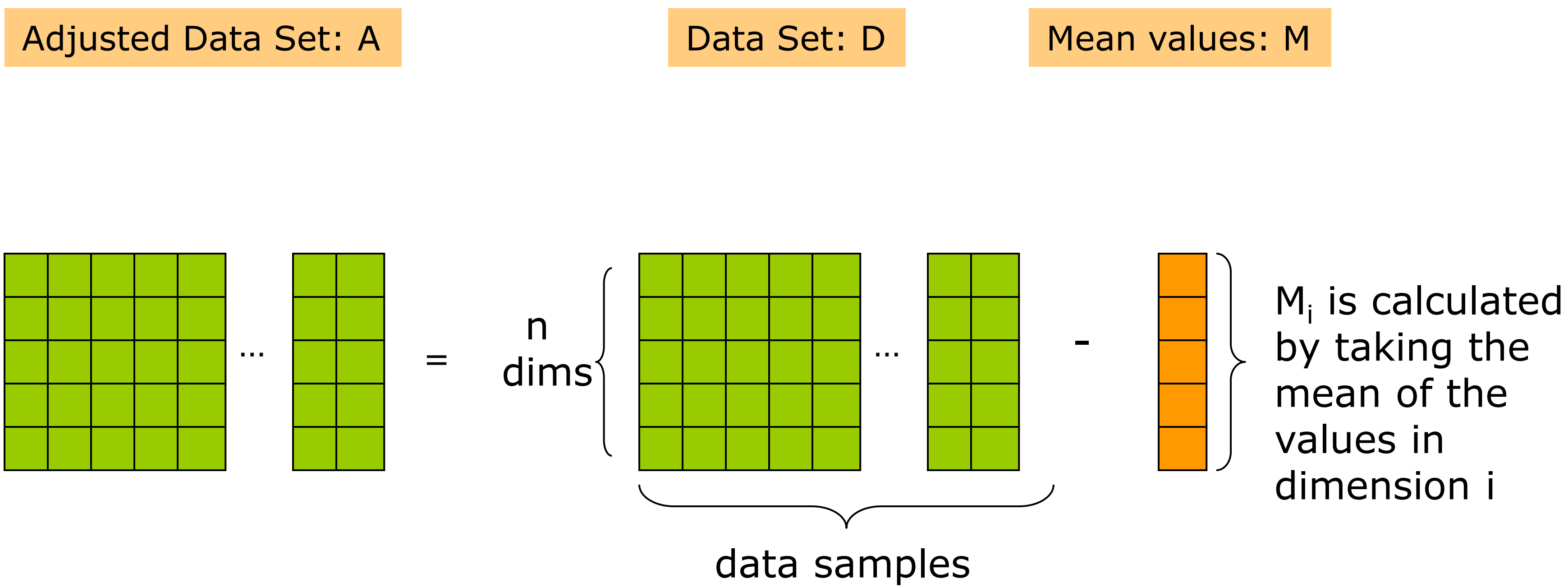
PCA gives us:

1. A measure of how each variable is associated with one another.
2. The directions in which the data are dispersed.
3. The relative importance of these different directions.

PCA Algorithm

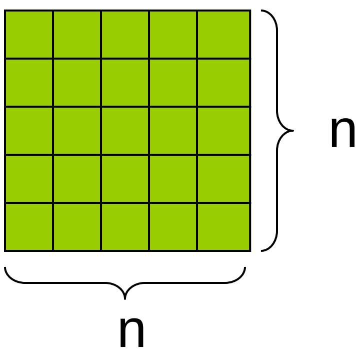
- PCA algorithm:
  - X ← Create N x d data matrix, with one row vector x_n per data point
  - X subtract mean *x* from each row vector x_n in X
  - Σ ← Find covariance matrix of X
  - Find eigenvectors and eigenvalues of Σ
  - PC's ← the M eigenvectors with largest eigenvalues

Steps in PCA: 1 Calculate Adjusted Data Set



Steps in PCA: 2 Calculate Co-Variance matrix

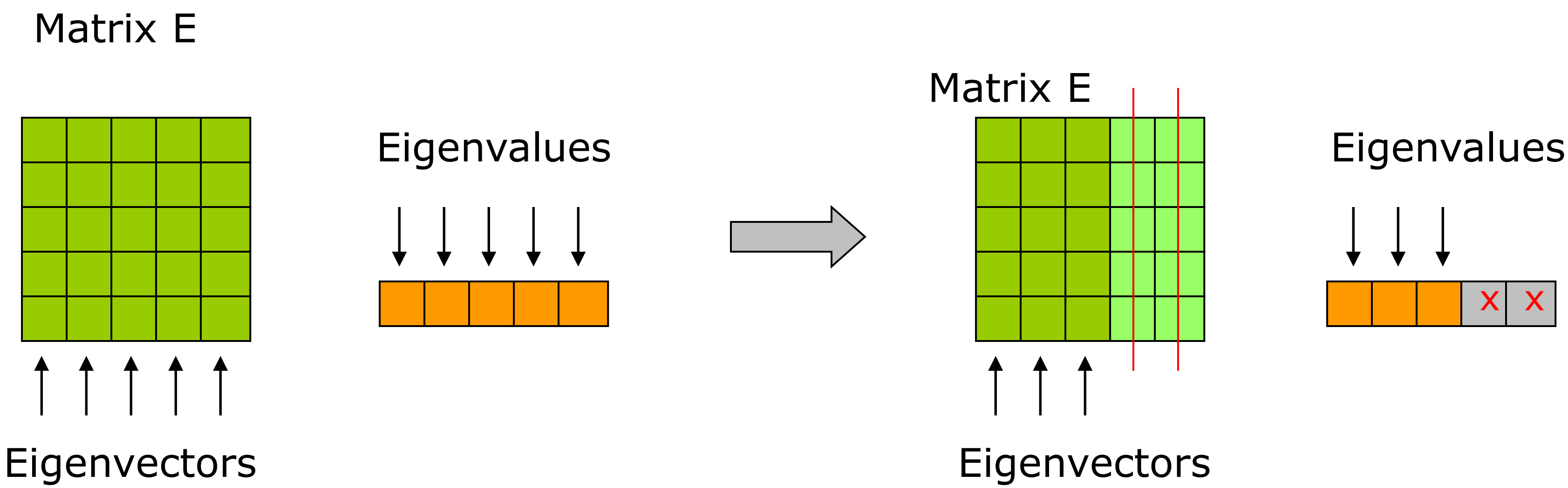
Co-variance Matrix: C



$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

$C_{ij} = cov(i,j)$

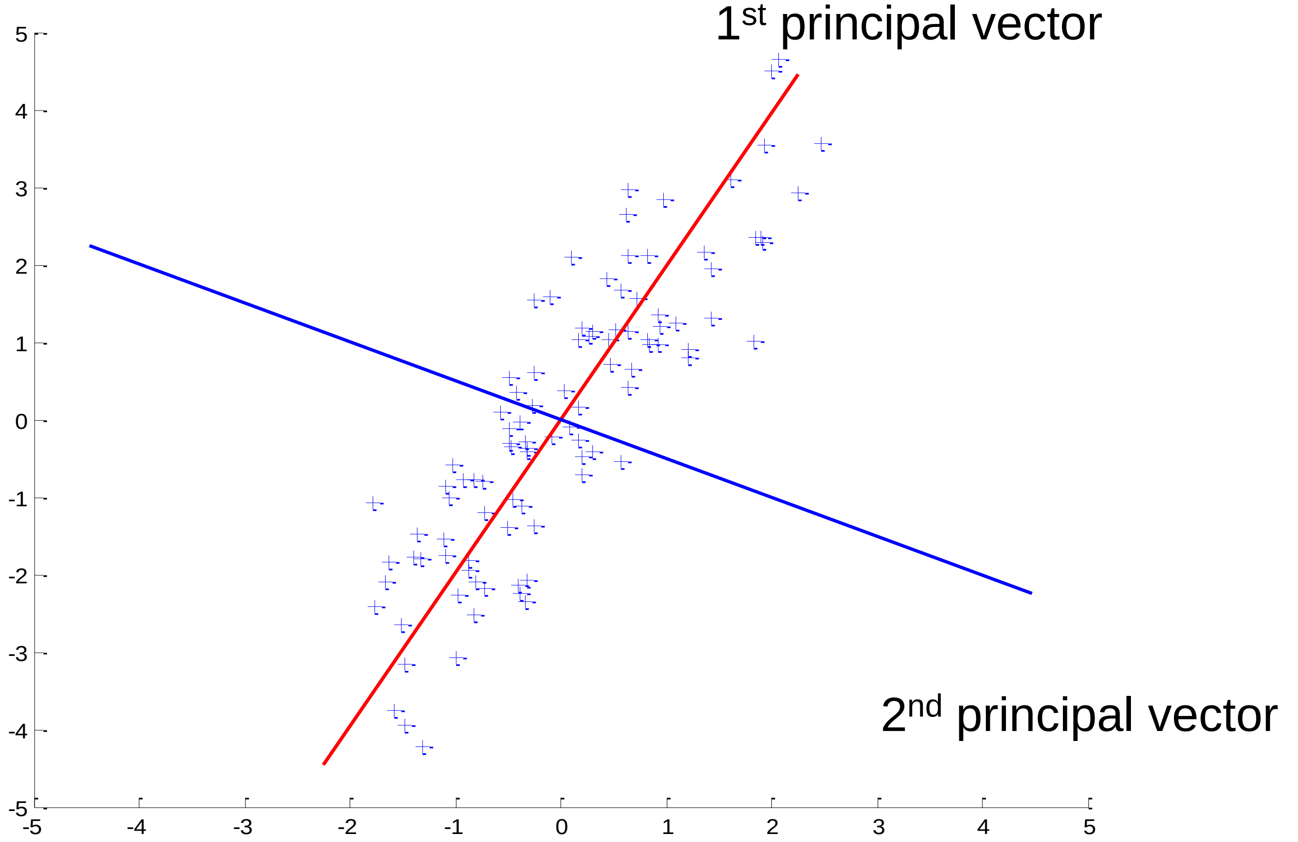
Steps in PCA: 3 Cal. Eigenvectors and Eigenvalues



If some eigenvalues are 0 or very small, we can essentially discard those eigenvalues and the corresponding eigenvectors, hence reducing the dimensionality of the new basis.

Principal Components

- **Gives best axis to project**
- **Minimum RMS error**
- **Principal vectors are orthogonal**



Eigenvalues & Eigenvectors

Definition: The ***eigenvalues*** of a real matrix **M** are the real numbers λ for which there is a nonzero vector **e** such that

$$\boxed{\mathbf{Me} = \lambda \mathbf{e}.}$$

The ***eigenvectors*** of **M** are the nonzero vectors **e** for which there is a real number λ such that **M**e = λ **e**.

If **M**e = λ **e** for **e** ≠ 0, then **e** is an ***eigenvector*** of **M** associated with ***eigenvalue*** λ, and vice versa. The eigenvectors and corresponding eigenvalues of **M** constitute the ***Eigen system*** of **M**.

To Calculate Eigen Vector and values

$$\mathbf{Det(A- \lambda I) =0}$$

Eigenvalues & Eigenvectors

Calculate the Determinant of the matrix

|A| means the determinant of the matrix A

First of all the matrix must be **square** (i.e. have the same number of rows as columns).

For a 2×2 Matrix

For a 2×2 matrix (2 rows and 2 columns):

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The determinant is:

$$|A| = ad - bc$$

"The determinant of A equals a times d minus b times c"

Eigenvalues & Eigenvectors

Calculate the Determinant of the matrix

For a 3×3 Matrix

For a 3×3 matrix (3 rows and 3 columns):

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

The determinant is:

$$|A| = a(ei - fh) - b(di - fg) + c(dh - eg)$$

"The determinant of A equals ... etc"

$$C = \begin{bmatrix} 6 & 1 & 1 \\ 4 & -2 & 5 \\ 2 & 8 & 7 \end{bmatrix}$$

$$\begin{aligned} |C| &= 6\times(-2\times7 - 5\times8) - 1\times(4\times7 - 5\times2) + 1\times(4\times8 - (-2\times2)) \\ &= 6\times(-54) - 1\times(18) + 1\times(36) \\ &= \mathbf{-306} \end{aligned}$$

Eigenvalues & Eigenvectors (Con't)

Example: Consider the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

It is easy to verify that $\mathbf{Me}_1 = \lambda_1 \mathbf{e}_1$ and $\mathbf{Me}_2 = \lambda_2 \mathbf{e}_2$ for $\lambda_1 = 1, \lambda_2 = 2$ and

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In other words, $\mathbf{e}_1$ is an eigenvector of **M** with associated eigenvalue λ_1, and similarly for $\mathbf{e}_2$ and λ_2.

Eigenvalues & Eigenvectors (Con't)

Example: Find the Eigen Value and Eigen Vector of the following Matrix

$$A=[4, 1; 3, 2]$$

Solution- Equation- **Det(A- λI) =0**

$$\begin{pmatrix} 4-\lambda & 1 \\ 3 & 2-\lambda \end{pmatrix} = 0$$

$$=(4-\lambda)\ (2-\lambda)-3=0$$

The characteristic equation is $\lambda^2-6\lambda+5=0$

So λ is 1 or 5.

When $\lambda=1$

$$\begin{pmatrix} 4-1 & 1 \\ 3 & 2-1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Eigenvalues & Eigenvectors (Con't)

The Equations obtained are

$$3x_1+x_2 = 0$$

And $$3x_1+x_2 = 0$$

$$x_2 = -3x_1$$

Many solutions are possible, but the simplest is x_1=1 and x_2=-3

So the first Eigen vector is [1,-3]

When λ=5 $$\begin{pmatrix} 4-5 & 1 \\ 3 & 2-5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The equations obtained are

$$-x_1 + x_2 = 0$$

$$3x_1 - 3\,x_2 = 0$$

$$x_2 = x_1$$

Many solutions are possible, but the simplest is x_1 = 1 and x_2 =1

So the Second Eigen vector is [1, 1]

PCA Algorithm

The steps involved in PCA Algorithm are as follows

Step-01: Get data.

Step-02: Compute the mean vector (μ).

Step-03: Subtract mean from the given data.

Step-04: Calculate the covariance matrix.

Step-05: Calculate the **eigen vectors** and **eigen values** of the **Covariance matrix.**

Step-06: Choosing components and forming a feature vector.

Step-07: Deriving the new data set.

Principal Components-Example

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

$\overline{X} = 1.81$

$\overline{Y} = 1.91$

$$C = \begin{pmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(Y,X) & Cov(Y,Y) \end{pmatrix}$$

$$C = \begin{pmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{pmatrix}$$

$$\mathbf{Cov(X,Y)} = \sum_{i=1}^{n} (Xi - \overline{X}).(Yi - \overline{Y})/(n-1)$$

X	$X-\overline{X}$	$(X-\overline{X})(X-\overline{X})$
2.5	0.69	0.476
.5	-1.31	1.7161
....	....	.....

Sum=5.5490

Y	$Y-\overline{Y}$	$(Y-\overline{Y})(Y-\overline{Y})$
2.4	0.49	0.2401
.7	-1.21	1.4641
....	....	......

Sum=6.449

X	Y	$X-\overline{X}$	$Y-\overline{Y}$	$(X-\overline{X})(Y-\overline{Y})$
2.5	2.4	0.69	0.49	0.3381
0.5	0.7	-1.31	-1.21	1.5851
...	...	.....	.....	......

Sum=5.539

Principal Components-Example

$|C-\lambda I| = 0$ **Equation To find eigen Values, C-Covariance matrix**

$$\begin{pmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0.6165 - \lambda & 0.6154 \\ 0.6154 & 0.7165 - \lambda \end{pmatrix}$$

$\lambda^2 -1.333\lambda+0.0630=0$ Solving the **Quadratic Equation**, we can find $\lambda 1$ & $\lambda 2$

$\lambda_1=0.0490$, $\boldsymbol{\lambda_2=1.2840}$

$CV=\lambda V$ **Equation To find eigen Vectors**

$$\begin{pmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{pmatrix} \begin{pmatrix} X1 \\ Y1 \end{pmatrix} = 0.0490 \begin{pmatrix} X1 \\ Y1 \end{pmatrix}$$

Numerical

Principal Components-Example

0.6165X1 +0.6154Y1= 0.0490 X1

0.6154 X1 + 0.7165 Y1= 0.0490 Y1

0.5674X1 = - 0.6154 Y1

0.6154 X1 = - 0.6674 Y1

$$\begin{pmatrix} -0.7351 \\ 0.6778 \end{pmatrix}$$

X2=0.92194 Y2

$$\begin{pmatrix} 0.92194 \\ 1 \end{pmatrix} = 0.8499+1$$

$$=\sqrt{1.8499}$$

=1.3601

X1= -1.0845 Y1 Assume Y1=1

$$\begin{pmatrix} -1.0845 \\ 1 \end{pmatrix} = 1.17614 +1$$

=SQRT (2.17614)

=1.47517

$$\begin{pmatrix} 0.6778 \\ 0.7351 \end{pmatrix}$$

SVD

Singular Value Decomposition

Unitary Matrix

- **What is Unitary Matrix**

A Matrix A is a Unitary Matrix if

$$A^{-1} = A^{*T}$$

A^* =Conjugate of A

A^{*T} = Transpose of Conjugate of A

Singular Value Decomposition

- The SVD for Square matrices was discovered independently by Beltrami in 1873 and Jordan in 1874, and extended to rectangular matrices by Eckart and Young in 1930s.
- The SVD of a rectangular matrix A is a decomposition of the form

$$A = U\,S\,V^T \qquad \text{OR} \qquad A = U\,S\,V^H$$

Where A is an m x n matrix.

U and V are orthogonal/ Unitary matrices.

S is a diagonal matrix comprised of singular values of A.

SVD - Properties

THEOREM : always possible to decompose matrix A into $A = U S V^T$, where

- U, S, V: unique
- U, V: column orthonormal (ie., columns are unit vectors, orthogonal to each other)
  - $U^TU = I$; $V^TV = I$ (I: identity matrix)
- S: singular value are positive, and sorted in decreasing order.

SVD Properties

- The Singular values $\sigma 1 \geq \sigma 2 \geq$--------- $\sigma n \geq 0$, appears in descending order along the main diagonal of S.
- The Singular values are obtained by taking the square root of the Eigen values of AA^T and A^TA.

$$\mathbf{A=USV^T}$$

$$\mathbf{A=[u_1,u_2,---------u_n]}\begin{bmatrix} \sigma 1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \sigma n \end{bmatrix}\begin{bmatrix} V_1^T \\ \\ Vn^T \end{bmatrix}$$

The Matrix V can be computed through the Eigen Vector of A^TA and the matrix U can be computed through the Eigen vectors of AA^T.

The rank of the matrix A is equal to the numbers of it's non zero singular values.

SVD - Properties

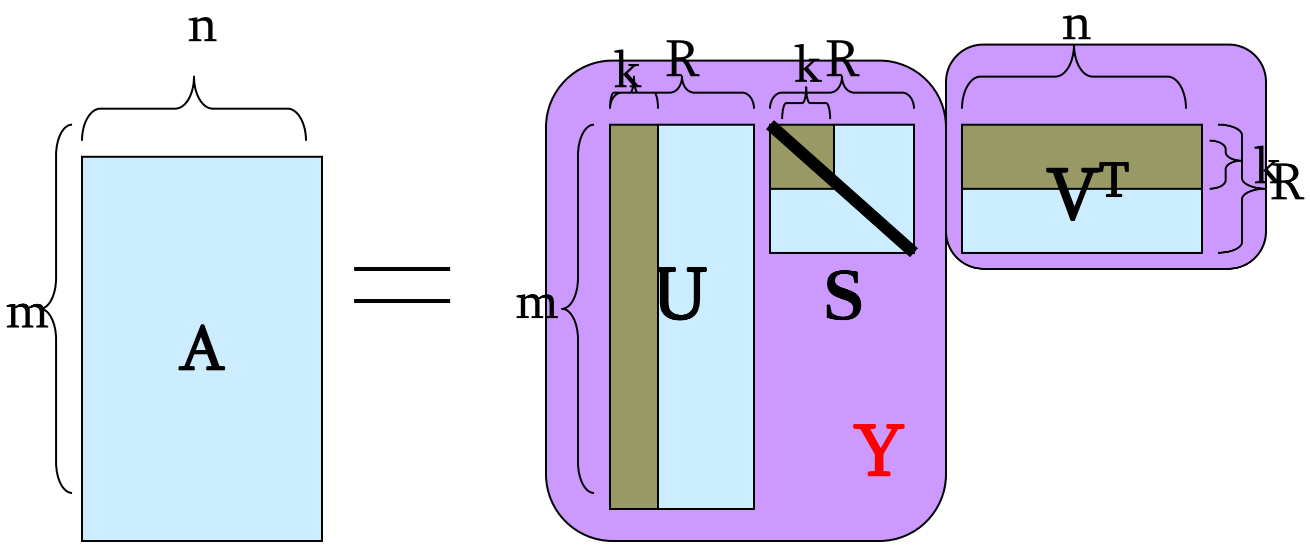
'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \text{ x } \begin{bmatrix} \lambda_1 & \emptyset \\ \emptyset & \lambda_2 \end{bmatrix} \text{ x } \begin{bmatrix} \text{—} & v_1 & \text{—} \\ \text{—} & v_2 & \text{—} \end{bmatrix}$$

Singular Value Decomposition

- **SVD**

$$A = U\,S\,V^T$$



- Best rank k approximation.
- PCA is an important application of SVD.

SVD - Example

- **A = U S V^T - example:**

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD – Dimensionality reduction

- **Q: How exactly is dim. reduction done?**
- **A: set the smallest singular values to zero:**

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
=
\begin{bmatrix}
0.18 & \cancel{0} \\
0.36 & \cancel{0} \\
0.18 & \cancel{0} \\
0.90 & \cancel{0} \\
0 & \cancel{0.53} \\
0 & \cancel{0.80} \\
0 & \cancel{0.27}
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & \cancel{5.29}
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0.71} & \cancel{0.71}
\end{bmatrix}
$$

SVD - Dimensionality Reduction

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}
$$

SVD - Dimensionality Reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Applications of SVD in Image Processing

- SVD approach can be used in Image Compression.
- SVD can be used in face recognition.
- SVD can be used in watermarking.
- SVD can be used for Texture classification.
- SVD can be used for Dynamic Texture Synthesis.

Kernel PCA

- Kernel PCA is an enhanced PCA method that incorporates a kernel function to determine principal components in different high-dimensional space, thereby facilitating solution of non-linear problems.
- KPCA finds new directions based on kernel matrix.
- KPCA is limited by an inability to determine importance of variables in contrast to linear PCA where it is possible to identify key variables that contribute to PCA score profiles.

Independent Component Analysis (ICA)

- ICA is a computational method for separating a multivariate signals into additive subcomponents.
- ICA works under the assumption that the subcomponents comprising the signal sources of the main signal are built from non-Gaussian sources.
- Moreover they are statistically independent from each other.
- ICA plays a dominant role in medical research in biomedical signal extraction and separation.
- Biomedical signals from many sources including hearts, brains and endocrine systems pose greatest challenge to researchers, radiologists for identifying each body part.
- Researchers need to separate weak signals arriving from multiple sources that is contaminated with artifacts and noise.

Disadvantages of Dimensionality Reduction

- Some data may be lost due to Dimensionality Reduction.
- In the PCA dimensionality reduction technique, sometimes the principal components required to consider are unknown.

Numericals-SVD

The Singular Value Decomposition: Let A be any $m \times n$ matrix. Then there are orthogonal matrices U, V and a diagonal matrix Σ such that

$$A = U\Sigma V^T$$

Specifically:

- The ordering of the vectors comes from the ordering of the singular values (largest to smallest).
- The columns of U are the eigenvectors of AA^T
- The columns of V are the eigenvectors of $A^T A$.
- The diagonal elements of Σ are the singular values, $\sigma_i = \sqrt{\lambda_i}$

Numericals-SVD

- Find the singular values of the matrix

$$B = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Solution. We use the same approach: $AA^T = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$. This has characteristic polynomial $\lambda^2 - 10\lambda + 9$, so $\lambda = 9$ and $\lambda = 1$ are the eigenvalues. Hence the singular values are 3 and 1.

Numericals-SVD

- Find the singular values of the matrix

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Solution. We compute AA^T. (This is the smaller of the two symmetric matrices associated with A.) We get $AA^T = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 2 \end{bmatrix}$. We next find the eigenvalues of this matrix. The characteristic polynomial is $\lambda^3 - 6\lambda^2 + 6\lambda = \lambda(\lambda^2 - 6\lambda + 6)$. This gives three eigenvalues: $\lambda = 3 + \sqrt{3}$, $\lambda = 3 - \sqrt{3}$ and $\lambda = 0$. Note that all are positive, and that there are two nonzero eigenvalues, corresponding to the fact that A has rank 2.

For the singular values of A, we now take the square roots of the eigenvalues of AA^T, so $\sigma_1 = \sqrt{3 + \sqrt{3}}$ and $\sigma_2 = \sqrt{3 - \sqrt{3}}$. (We don't have to mention the singular values which are zero.)

Numericals-SVD

- Find the singular values of the matrix

$$A = \begin{bmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

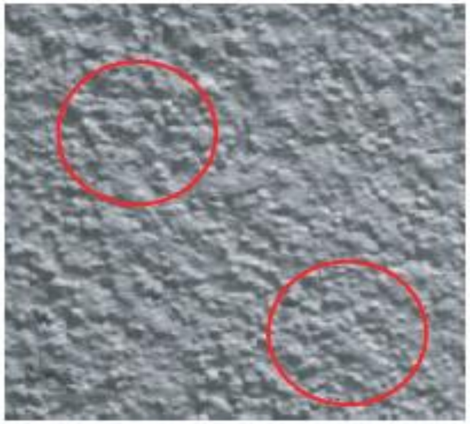
Solution. We compute AA^T and find $AA^T = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 6 & 2 \\ 2 & 2 & 2 \end{bmatrix}$. The characteristic polynomial is

$$\begin{aligned} -\lambda^3 + 10\lambda^2 - 16\lambda &= -\lambda(\lambda^2 - 10\lambda + 16) \\ &= -\lambda(\lambda - 8)(\lambda - 2) \end{aligned}$$

So the eigenvalues of AA^T are $\lambda = 8, \lambda = 2, \lambda = 0$. Thus the singular values are $\sigma_1 = 2\sqrt{2}, \sigma_2 = \sqrt{2}$ (and $\sigma_3 = 0$).

Introduction

- Texture-repetition of a specific structure, which is not limited only to the visual domain.



Example of a texture image (left) and a natural image (right).

Texture Types

- **Static Texture-** as an image showing spatial stationarity.
- **Dynamic Texture-** a sequence of images showing temporal stationarity.

Static Texture

Dynamic Texture

Dynamic Texture Synthesis

Synthesis- Definition, Different approach

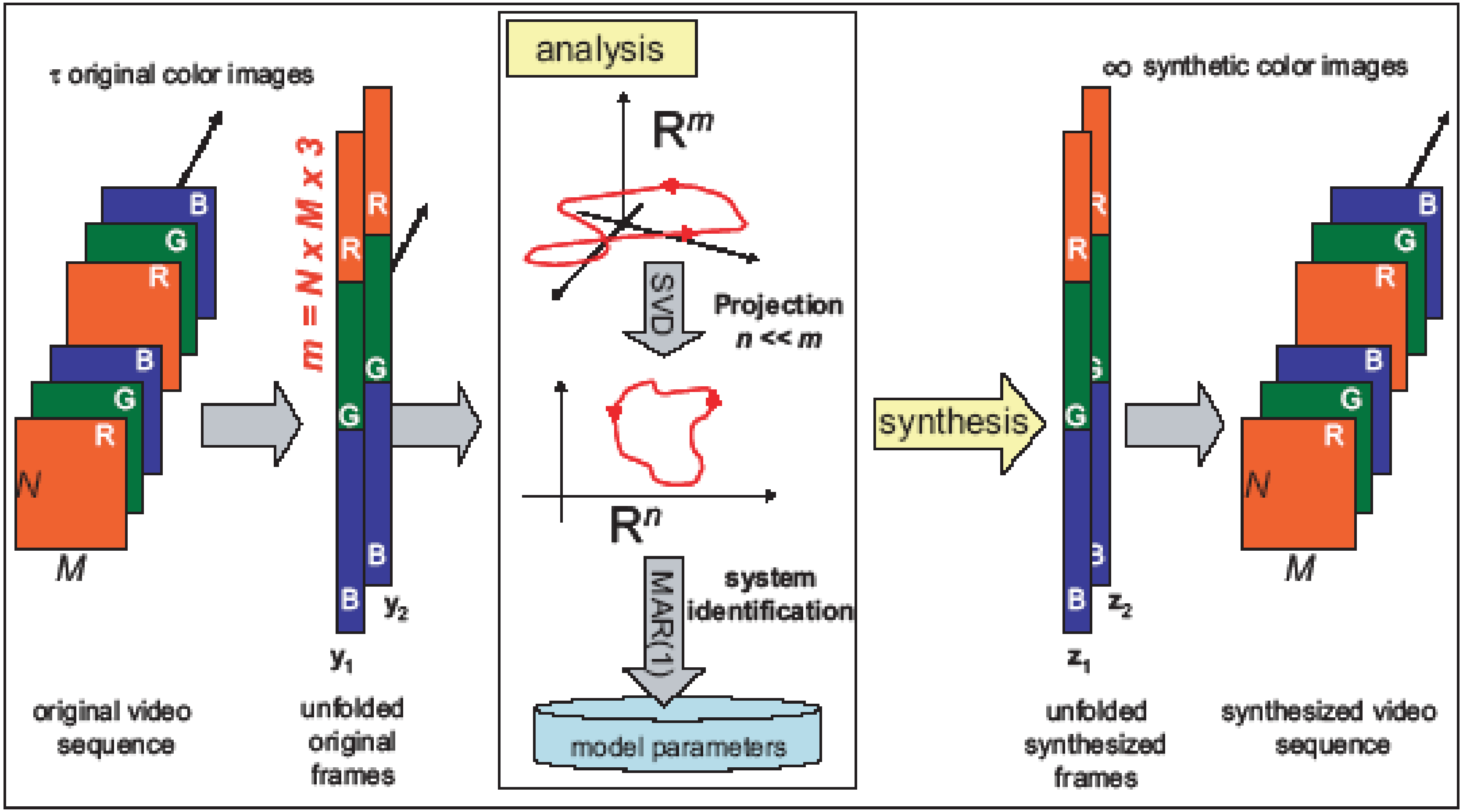
- **Physics-Based Approach- Natural phenomena, High quality, Very flexible, Specific to particular texture, Costly (Rendering Technique).**
- **Image Based Approach –Two Types**
  - **Non-Parametric Approach**- Extracts different video clips from the original video and patching them together to obtain a new video, High quality, Not Flexible, On the fly synthesis is not possible.
  - **Parametric Approach**-Parametric techniques are based on a model of the dynamic texture. Compact representation, More Flexible. On-the fly synthesis is possible.

Comparison

Between different approaches for Dynamic Texture Synthesis

Different Parameters	Physics-based	Image based	
		Patch	**Parametric**
Analysis Cost	HIGH	LOW	LOW
Synthesis Cost	HIGH	LOW	LOW
Model Size	LOW	HIGH	LOW
Specificity	HIGH	LOW	LOW
Flexibility	HIGH	LOW	HIGH

SVD-Soatto-Doretto Model



Soatto-Doretto's model for the analysis and synthesis of dynamic textures.

Analysis

The analysis consists in -

1) Finding an appropriate subspace to represent the trajectory.
2) Identifying the trajectory.

The analysis is done as follows.

First step-

A sequence of τ color images of size N ×M is reshaped into a sequence of column vectors of length m = N × M ×3. The original colour encoding is RGB. The column vectors are ordered in matrix

$$\mathbf{Y}_1^\tau = [\mathbf{y}_1, \ldots, \mathbf{y}_\tau] \in \mathbb{R}^{m \times \tau}$$

The temporal mean of each pixel is computed stored in a vector d $\in \mathbb{R}^m$ and subtracted from $\mathbf{Y}_1^\tau$ Then obtained a new matrix by using SVD,

$$\bar{\mathbf{Y}}_1^\tau = \mathbf{USV}^T$$

where $\mathbf{U} \in \mathbb{R}^{m \times \tau}$, $\mathbf{V} \in \mathbb{R}^{\tau \times \tau}$, and $\mathbf{S} \in \mathbb{R}^{\tau \times \tau}$.

Analysis

The columns of U and V are defined as the left and right singular vectors, respectively, and S is a diagonal matrix formed by the τ singular values of $\bar{\mathbf{Y}}_1^\tau$ ordered according decreasing energy.

Dimension reduction is performed by retaining the first n singular values of the decomposition. Indicating with U_n and V_n the matrices collecting the first n columns of U and V, and with S_n the diagonal matrix collecting the first n diagonal elements of S, the matrix

$$\hat{\bar{Y}}_1^\tau = [\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_\tau] = \mathbf{U}_n \mathbf{S}_n \mathbf{V}_n^T$$

Above Equation defines the best n-rank approximation of matrix $\bar{\mathbf{Y}}_1^\tau$,

Assume that $C = U_n$ and $X = S_n V_n^T$

Analysis

The generic column vector $\bar{y}_k$ is thus: $\bar{\mathrm{y}}_k = \mathbf{C}\mathbf{x}_k + \mathbf{e}$

where x_k is the k-th column of matrix X and e is the residual error. Written differently, we see that

$$\bar{\mathrm{y}}_k = \mathrm{c}_1 x_{1k} + \mathrm{c}_2 x_{2k} + \ldots + \mathrm{c}_n x_{nk} + \mathrm{e}$$

where c_i with i = 1, . . . , n indicates the i-th column vectors of the matrix C. Above Equation shows that $\bar{y}_k$ is a linear combination of the columns of matrix C, which contains the basis of the subspace to which image vectors y are projected.

As an example, in the case of the video “Flame”, the columns of matrix C contain the “Eigen-flames” whose linear combination generates one flame image.

Analysis

Matrix X fixes the weights of the linear combination, thus establishing how frames change in time. In other words, it represents the dynamics of the video sequence.

The second step is the identification of the system dynamics. This is done starting from the matrix $\mathbf{X} \in \mathbb{R}^{n \times \tau}$ which is associated to the time evolution of the video frame

A MAR(1) model is used to represent the evolution of x_k

$$\mathrm{x}_k = \mathrm{A}\mathrm{x}_{k-1} + \mathrm{B}\mathrm{v}_k$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times n_v}$ are matrices computed from $\mathbf{X}$

vk is a Gaussian random noise vector i.e. correlation of the noise in time

MAR(1)-Multivariate Autoregressive model-It consists of two or more variable quantities and it used to predict future values based on past values.

Synthesis

Second Step-

The final model is thus represented by the following system of equations:

$$\mathbf{x_{k+1} = Ax_k + Bv_k}$$

$$\mathbf{z_k = Cx_k + d.}$$

Where Matrix X- fixes the weights of the linear combination, thus Establishing how frames changes in time, and are $A \in R^{nxn}$ $B \in R^{nxnv}$ matrices computed from X. v_k is a Gaussian random noise vector


Diolch
Kiitos
umesc
Sheun
Dekuju/Dekujeme
Kasih
Todah
Mamnoon
Shnorhakalutiun
Teşekkür
Hvala
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ačí
Gamsahapnida
Ngiyabonga
Xie
Dank
Takk
Cam
Gra
or
Dakujem
Waad
Dhanyavaadalu
Kop
Salamat
Merci
al
Grazie
Faleminderit
Daw
Dhanyavaadalu
Dankie
Dhanyavaad
Go
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Shukriya
Or
Dhonnobaad
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Euxaristo
Kun
Hain
Dhan
Fyrir
Terima
Enkosi
Danke
dank
daa

Principal Components

$$C - \lambda I = 0$$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{bmatrix}$$

$$\rightarrow \lambda^2 - 1.333\lambda + 0.0630 = 0$$

$$\lambda_1 = 0.0490$$

$$\lambda_2 = 1.2840$$

$$CV = \lambda V$$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 0.0490 \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

$$0.6165 X_1 + 0.6154 Y_1 = 0.0490 X_1$$

$$0.6154 X_1 + 0.7165 Y_1 = 0.0490 Y_1$$

$$\rightarrow 0.5674 X_1 = -0.6154 Y_1$$

$$\rightarrow 0.6154 X_1 = -0.6674 Y_1$$

$$X_1 = -1.0845 Y_1$$

$$\begin{bmatrix} -1.0845 \\ 1 \end{bmatrix} \quad = 1.17614 + 1 = \sqrt{2.17614} = 1.47517$$

$$\Rightarrow \begin{bmatrix} -0.7351 \\ 0.6778 \end{bmatrix}$$

$$X_2 = 0.92194 Y_2$$

$$\rightarrow \begin{bmatrix} 0.92194 \\ 1 \end{bmatrix} \quad = 0.8499 + 1 = \sqrt{1.8499} = 1.3601$$

$$\Rightarrow \begin{bmatrix} 0.6778 \\ 0.7351 \end{bmatrix}$$