Dimensionality Reduction
Principal Component Analysis

Karl Pearson

Dr. Amani RAAD
ULFG1

amani.raad@ul.edu.lb
2022-2023

Motivation

- Dimensionality reduction
  - Another way to simplify complex high-dimensional data
  - Summarize data with a lower dimensional real valued vector

Motivation

- Dimensionality reduction
  - Another way to simplify complex high-dimensional data
  - Summarize data with a lower dimensional real valued vector

- Given data points in n dimensions
- Convert them to data points in $k<n$ dimensions
- With minimal loss of information

Main benefits

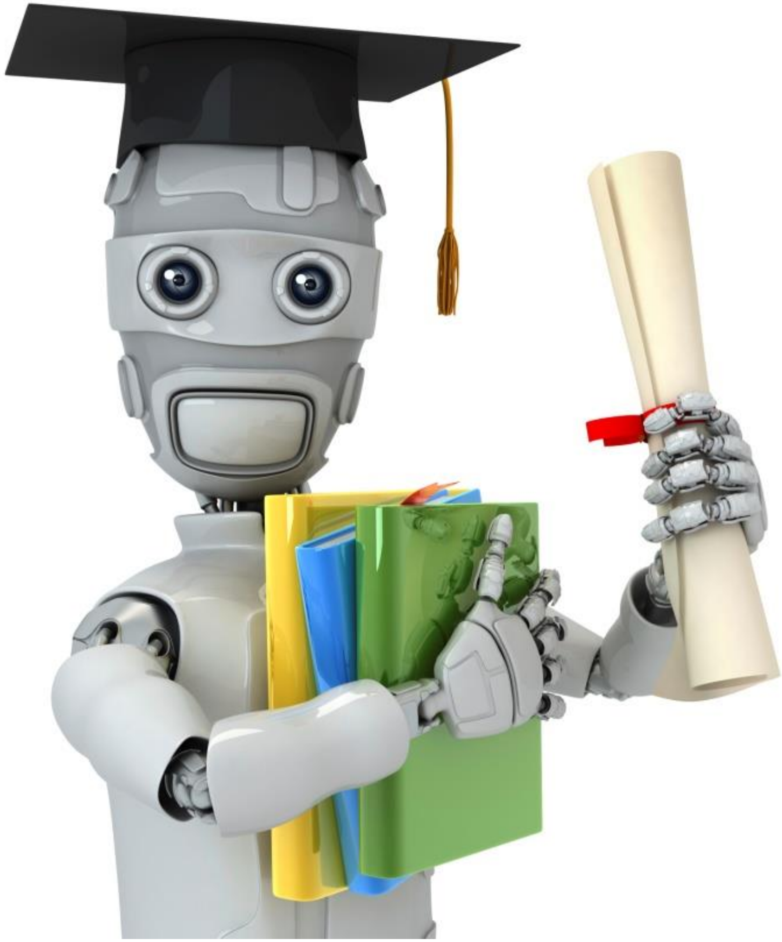
Dimension reduction offers several benefits such as-

- It compresses the data and thus reduces the storage space requirements.
- It reduces the time required for computation since less dimensions require less computation.
- It eliminates the redundant features.
- It improves the model performance.

Dimensionality reduction

- PCA (Principal Component Analysis):
  - Find projection that maximize the variance
- ICA (Independent Component Analysis):
  - Very similar to PCA except that it assumes non-Gaussian features
- Multidimensional Scaling:
  - Find projection that best preserves inter-point distances
- LDA(Linear Discriminant Analysis):
  - Maximizing the component axes for class-separation
- …
- …

Principal Component Analysis



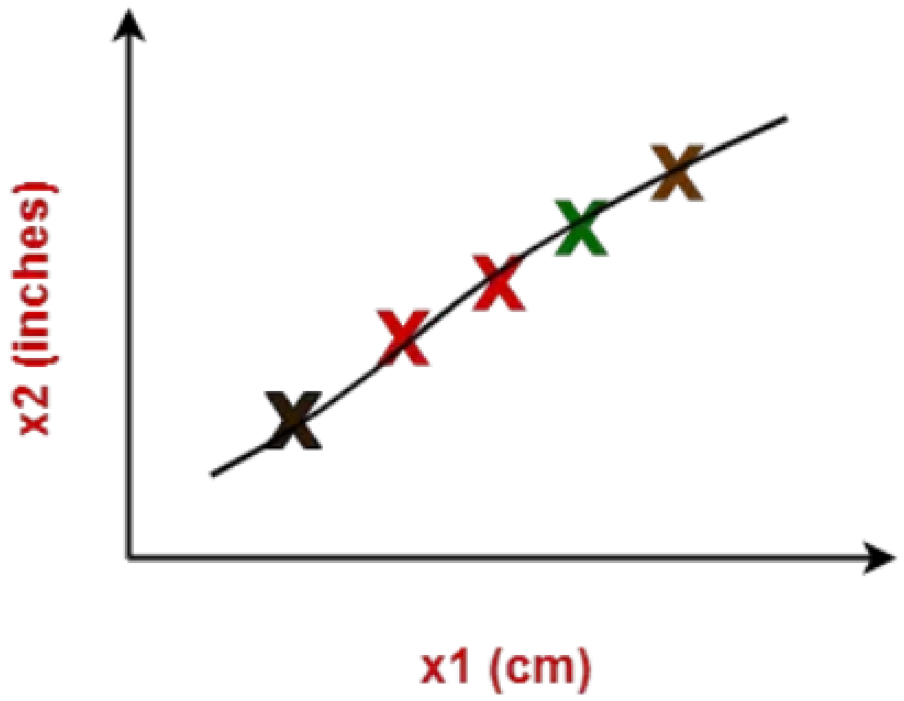
Dimensionality Reduction

Motivation I: Data Compression

Machine Learning

Consider the following example-

- The following graph shows two dimensions x1 and x2.
- x1 represents the measurement of several objects in cm.
- x2 represents the measurement of several objects in inches.



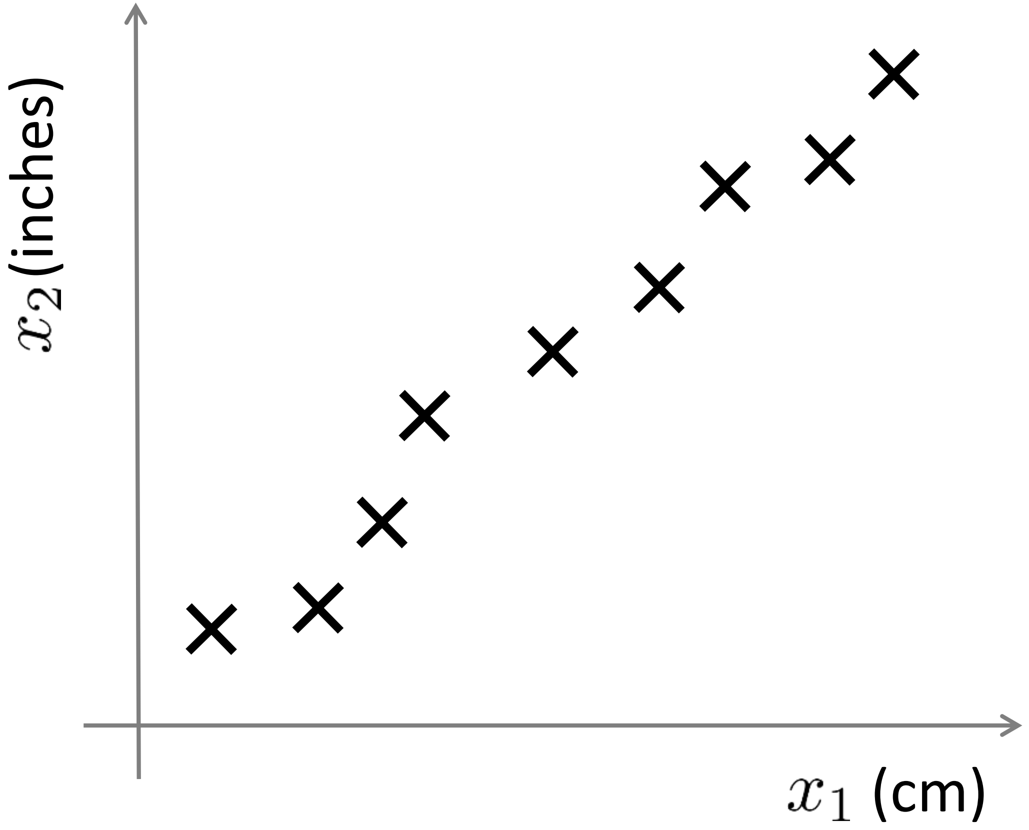
In machine learning,

- Using both these dimensions convey similar information.
- Also, they introduce a lot of noise in the system.
- So, it is better to use just one dimension.

Using dimension reduction techniques-

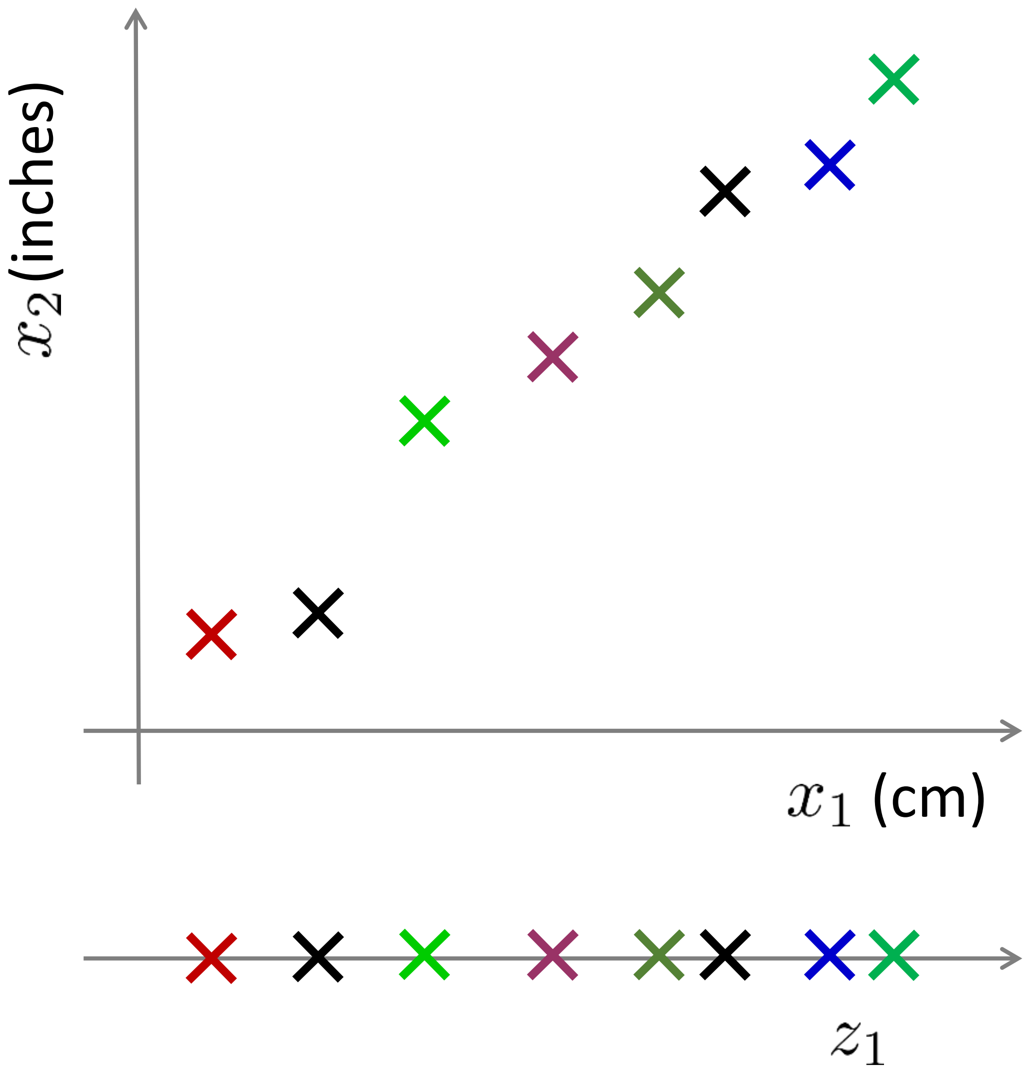
- We convert the dimensions of data from 2 dimensions ($x1$ and $x2$) to 1 dimension ($z1$).
- It makes the data relatively easier to explain.

Data Compression



Reduce data from 2D to 1D

Data Compression



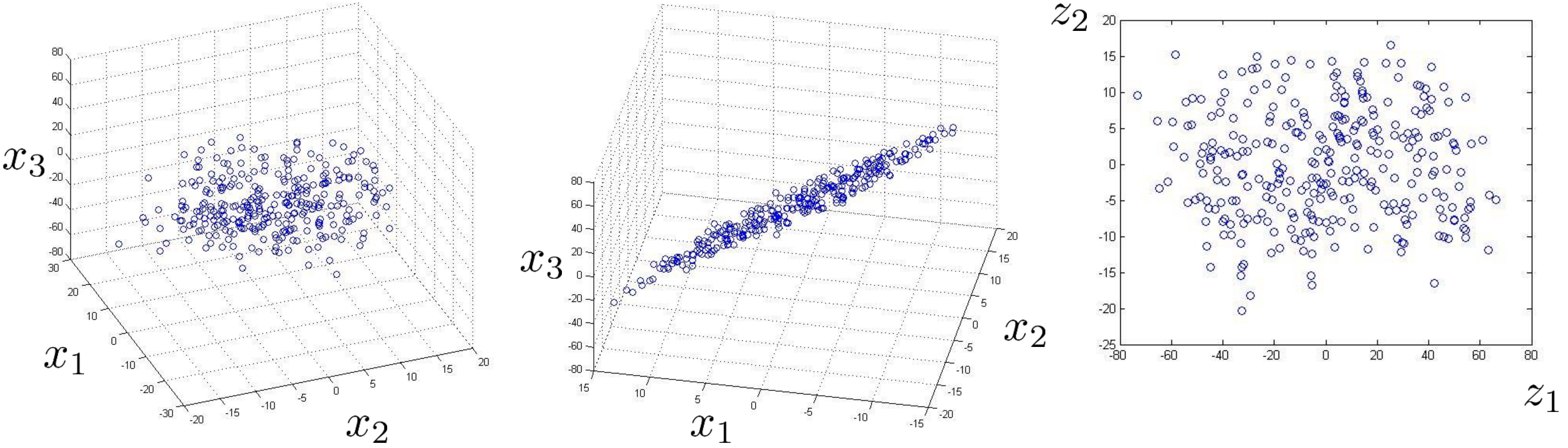
Reduce data from 2D to 1D

$$
\begin{aligned}
x^{(1)} &\rightarrow z^{(1)} \\
x^{(2)} &\rightarrow z^{(2)} \\
&\vdots \\
x^{(m)} &\rightarrow z^{(m)}
\end{aligned}
$$

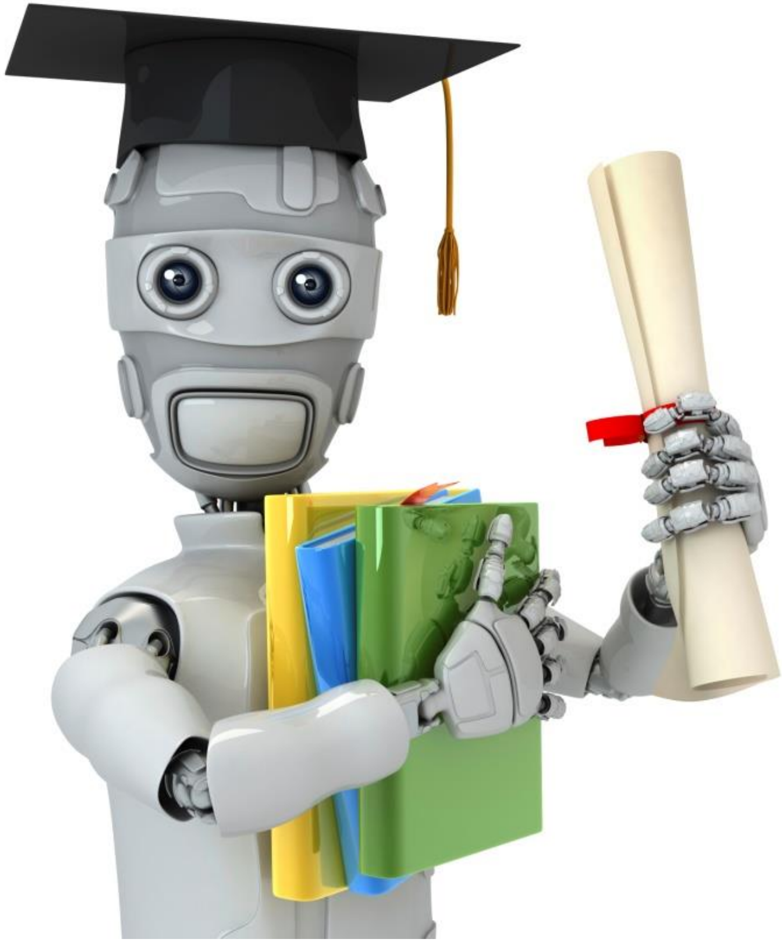
Andrew Ng

Data Compression

Reduce data from 3D to 2D



Andrew Ng



Dimensionality Reduction

Motivation II: Data Visualization

Machine Learning

Data Visualization

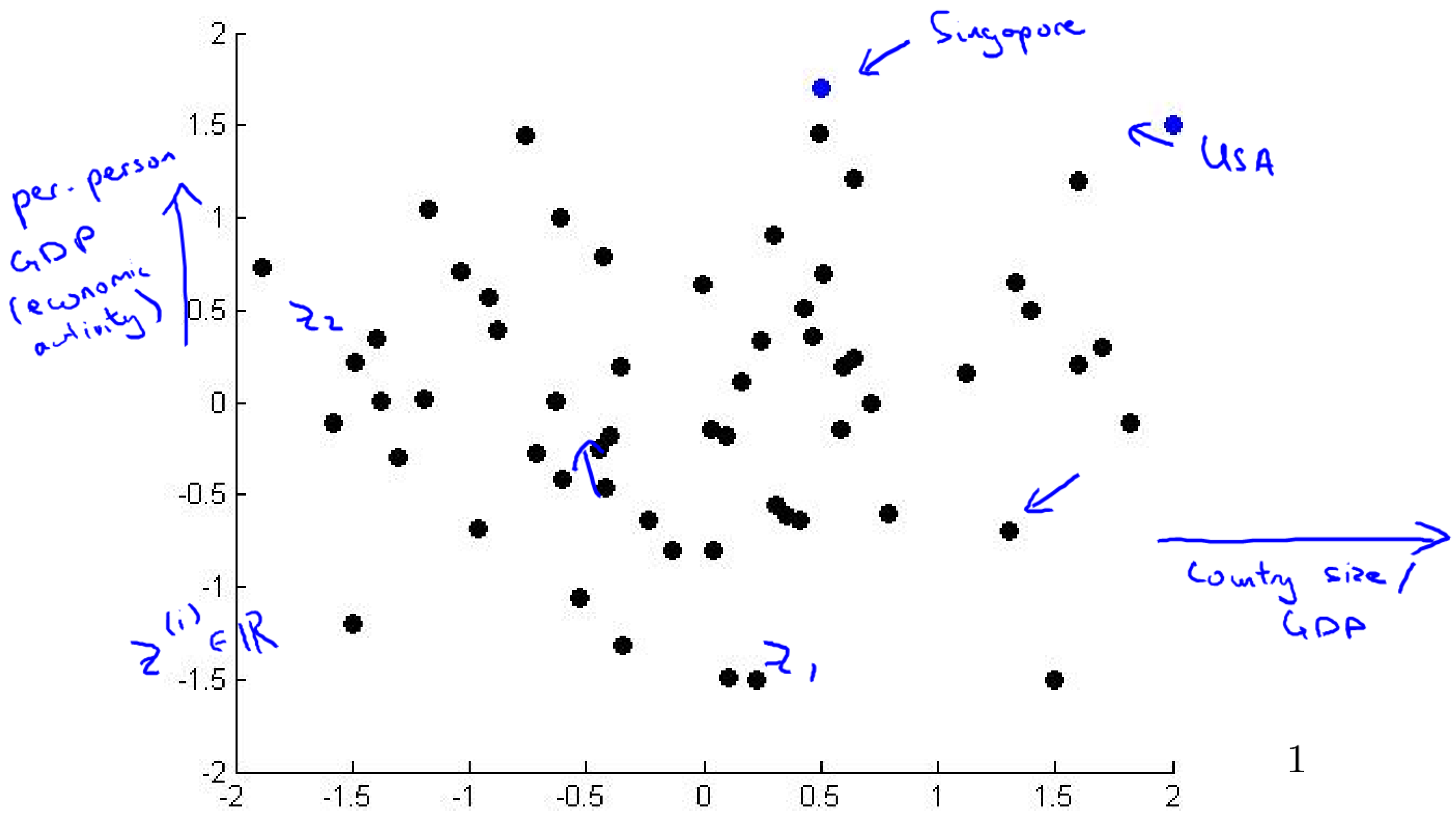
Country	GDP (trillions of US$)	Per capita GDP (thousands of intl. $)	Human Develop-ment Index	Life expectanc y	Poverty Index (Gini as percentage)	Mean household income (thousands of US$)	…
Canada	1.577	39.17	0.908	80.7	32.6	67.293	…
China	5.878	7.54	0.687	73	46.9	10.22	…
India	1.632	3.41	0.547	64.7	36.8	0.735	…
Russia	1.48	19.84	0.755	65.5	39.9	0.72	…
Singapore	0.223	56.69	0.866	80	42.5	67.1	…
USA	14.527	46.86	0.91	78.3	40.8	84.3	…
…	…	…	…	…	…	…	

[resources from en.wikipedia.org]

Data Visualization

Country	z_1	z_2
Canada	1.6	1.2
China	1.7	0.3
India	1.6	0.2
Russia	1.4	0.5
Singapore	0.5	1.7
USA	2	1.5
…	…	…

Data Visualization



Covariance

- Variance and Covariance:
  - Measure of the “spread” of a set of points around their center of mass(mean)
- Variance:
  - Measure of the deviation from the mean for points in one dimension
- Covariance:
  - Measure of how much each of the dimensions vary from the mean with **respect to each other**

- **Covariance is measured between two dimensions**
- **Covariance sees if there is a relation between two dimensions**
- **Covariance between one dimension is the variance**

one attribute first

- Question: how much spread is in the data along the axis? (distance to the mean)
- Variance=Standard deviation^2

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

Temperature
42
40
24
30
15
18
15
30
15
30
35
30
40
30

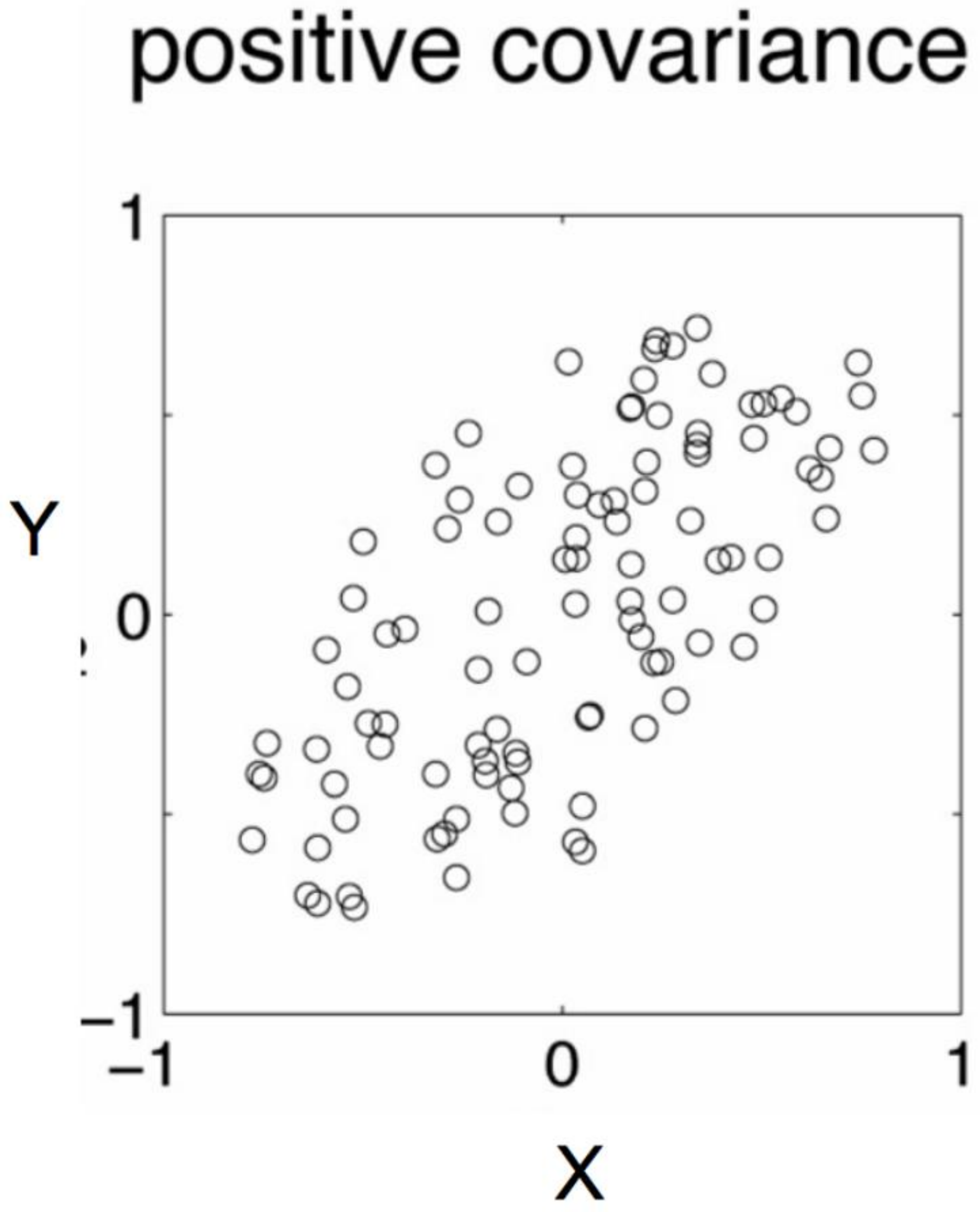
Now consider two dimensions

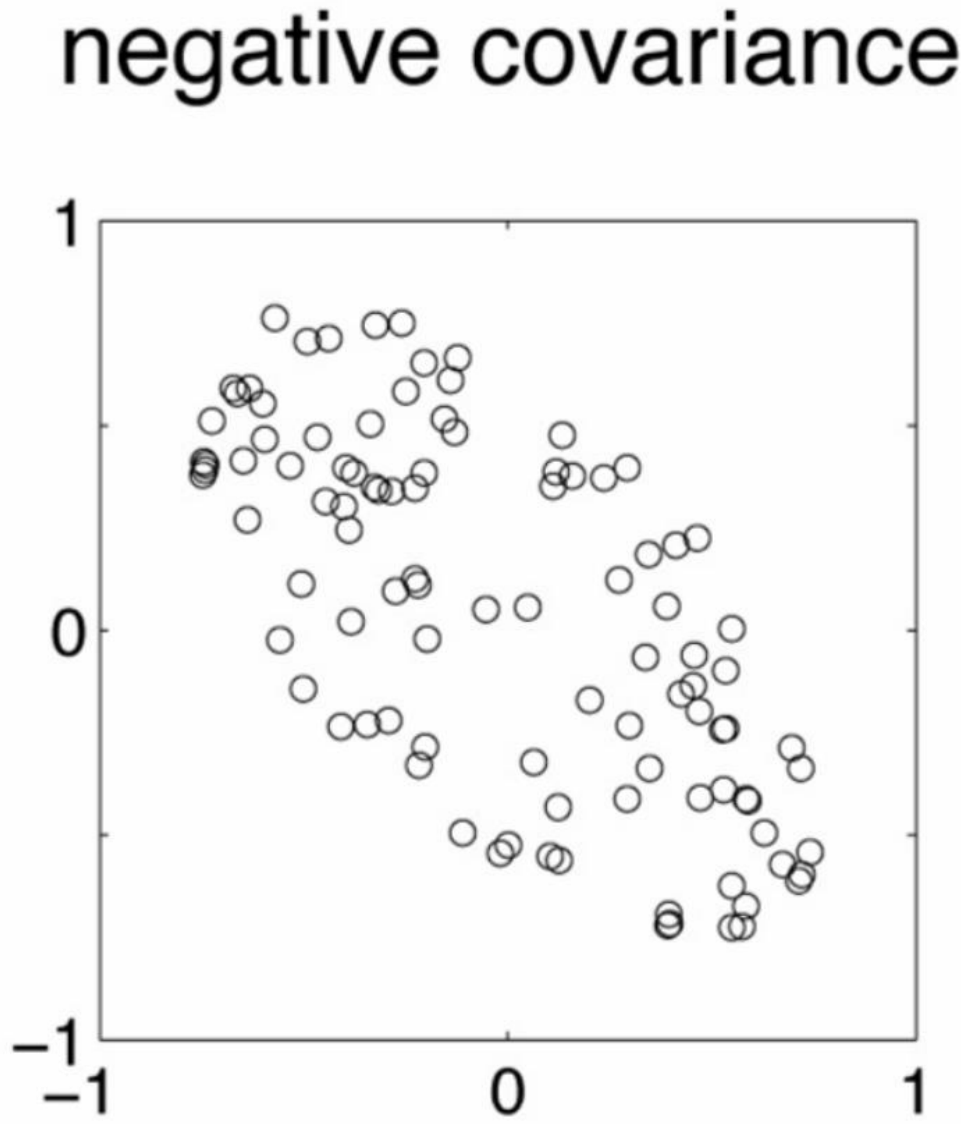
X=Temperature	Y=Humidity
40	90
40	90
40	90
30	90
15	70
15	70
15	70
30	90
15	70
30	70
30	70
30	90
40	70
30	90

Covariance: measures the correlation between X and Y

- cov(X,Y)=0: independent
- Cov(X,Y)>0: move same dir
- Cov(X,Y)<0: move oppo dir

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





Positive: Both dimensions increase or decrease together

Negative: While one increase the other decrease

More than two attributes: covariance matrix

- Contains covariance values between all possible dimensions (=attributes):

$$C^{nxn} = (c_{ij} \mid c_{ij} = \mathrm{cov}(Dim_i, Dim_j))$$

- Example for three attributes (x,y,z):

$$C = \begin{pmatrix} \mathrm{var}(x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{var}(y) & \mathrm{cov}(y,z) \\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{var}(z) \end{pmatrix}$$

Eigenvalues & eigenvectors

- Vectors **x** having same direction as *A***x** are called *eigenvectors* of *A* (*A* is an n by n matrix).
- In the equation *A***x**=λ**x**, λ is called an *eigenvalue* of *A*.

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} x \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4x \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Eigenvector and Eigenvalue

Ax = λx

A: Square Matrix

λ: Eigenvector or characteristic vector

X: Eigenvalue or characteristic value

Example

$$Show\ x = \begin{bmatrix} 2 \\ 1 \end{bmatrix} is\, an\, eigenvector\ for\ A = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix}$$

$$Solution : Ax = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$But\ \ for\ \ \lambda = 0,\ \ \lambda x = 0\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$Thus, x\, is\, an\, eigenvector\, of\ A, and\, \lambda = 0\ is\, an\, eigenvalue.$$

Eigenvector and Eigenvalue

$Ax = \lambda x$ → $Ax - \lambda x = 0$

$(A - \lambda I)x = 0$

If we define a new matrix B: → $B = A - \lambda I$

$Bx = 0$

If B has an inverse: → $x = B^{-1}0 = 0$ ✗ **BUT! an eigenvector cannot be zero!!**

x will be an eigenvector of A if and only if B does not have an inverse, or equivalently det(B)=0 :

$$\boxed{\det(A - \lambda I) = 0}$$

Eigenvector and Eigenvalue

Example 1: Find the eigenvalues of

$$A=\begin{bmatrix}2 & -12\\ 1 & -5\end{bmatrix}$$

$$|\lambda I - A| = \begin{vmatrix}\lambda-2 & 12\\ -1 & \lambda+5\end{vmatrix} = (\lambda-2)(\lambda+5)+12$$

$$=\lambda^2+3\lambda+2=(\lambda+1)(\lambda+2)$$

two eigenvalues: –1, – 2

Note: The roots of the characteristic equation can be repeated. That is, $\lambda_1 = \lambda_2 = \ldots = \lambda_k$. If that happens, the eigenvalue is said to be of multiplicity k.

Example 2: Find the eigenvalues of

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

$$|\lambda I - A| = \begin{vmatrix}\lambda-2 & -1 & 0\\ 0 & \lambda-2 & 0\\ 0 & 0 & \lambda-2\end{vmatrix} = (\lambda-2)^3=0$$

$\lambda = 2$ is an eigenvalue of multiplicity 3.

Eigenvalues & eigenvectors

- $A\mathbf{x}=\lambda\mathbf{x} \Leftrightarrow (A-\lambda \mathrm{I})\mathbf{x}=0$
- How to calculate **x** and λ:
  - Calculate $det(A-\lambda I)$, yields a polynomial (degree n)
  - Determine roots to $det(A-\lambda I)=0$, roots are eigenvalues λ
  - Solve $(A- \lambda I)\ \mathbf{x}=0$ for each λ to obtain eigenvectors **x**

Principal components

- 1. principal component (PC1)
  - The eigenvalue with the largest absolute value will indicate that the data have the largest variance along its eigenvector, the direction along which there is the biggest variation
- 2. principal component (PC2)
  - the direction with maximum variation left in data, orthogonal to the PC
- In general, only few directions manage to capture most of the variability in the data.

Remarks

- Principal components represent the directions of the data that explain a **maximal amount of variance**, that is to say, the lines that capture most information of the data.
- The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more information it has.
- Just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible

Transformed Data

- Eigenvalues λ_j corresponds to variance on each component *j*
- *Thus, sort by* λ_j
- Take the first top *p* eigenvectors
- These are the directions with the largest variances

Principal Component Analysis

Goal: Find r-dim projection that best preserves variance

1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

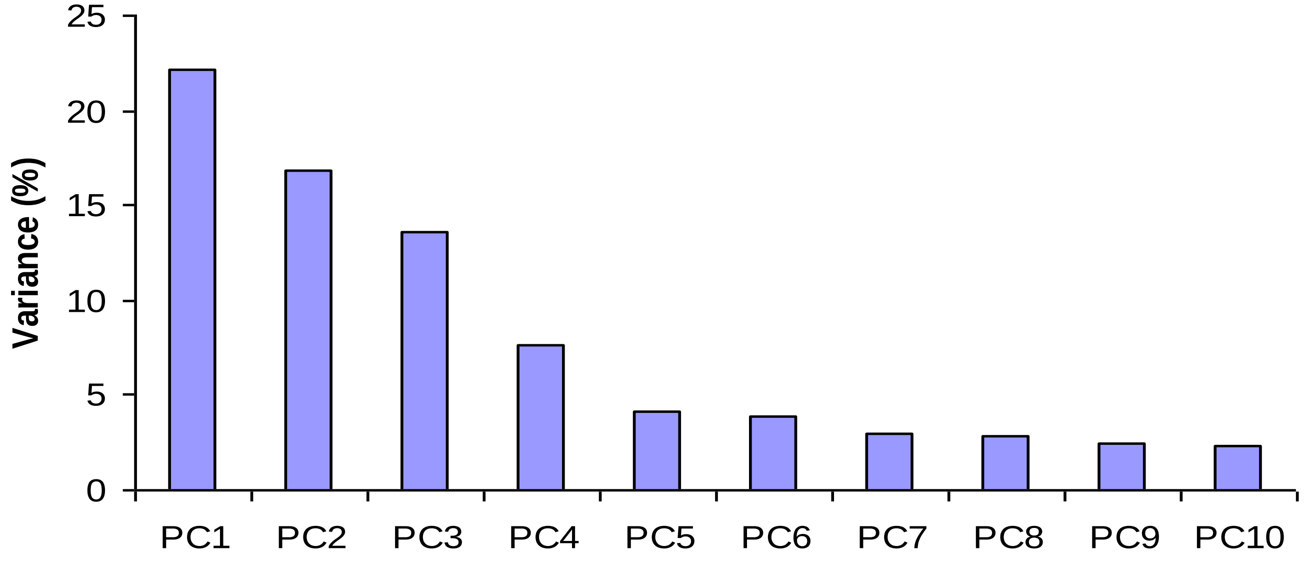
   where y is the new point, x is the old one, and the rows of A are the eigenvectors

Eigenvalues

- Calculate eigenvalues λ and eigenvectors **x** for covariance matrix:
  - Eigenvalues λ_j are used for calculation of [% of total variance] (V_j) for each component *j*:

$$V_j = 100 \cdot \frac{\lambda_j}{\sum_{x=1}^{n} \lambda_x}$$

Principal components - Variance



Example 1

Step-01:

Get data.

The given feature vectors are-

- x_1 = (2, 1)
- x_2 = (3, 5)
- x_3 = (4, 3)
- x_4 = (5, 6)
- x_5 = (6, 7)
- x_6 = (7, 8)

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} \begin{bmatrix} 6 \\ 7 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Step-02:

Calculate the mean vector (μ).

Mean vector (μ)

= ((2 + 3 + 4 + 5 + 6 + 7) / 6, (1 + 5 + 3 + 6 + 7 + 8) / 6)

= (4.5, 5)

Thus,

$$\text{Mean vector } (\mu) = \begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$$

Step-03:

Subtract mean vector (μ) from the given feature vectors.

- $x_1 - \mu = (2 - 4.5, 1 - 5) = (-2.5, -4)$
- $x_2 - \mu = (3 - 4.5, 5 - 5) = (-1.5, 0)$
- $x_3 - \mu = (4 - 4.5, 3 - 5) = (-0.5, -2)$
- $x_4 - \mu = (5 - 4.5, 6 - 5) = (0.5, 1)$
- $x_5 - \mu = (6 - 4.5, 7 - 5) = (1.5, 2)$
- $x_6 - \mu = (7 - 4.5, 8 - 5) = (2.5, 3)$

Feature vectors (x_i) after subtracting mean vector (μ) are-

$$\begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}$$

Calculate the covariance matrix.

Covariance matrix is given by-

$$\text{Covariance Matrix} = \frac{\Sigma (x_i - \mu)(x_i - \mu)^t}{n}$$

$$m_1 = (x_1 - \mu)(x_1 - \mu)^t = \begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -2.5 & -4 \end{bmatrix} = \begin{bmatrix} 6.25 & 10 \\ 10 & 16 \end{bmatrix}$$

$$m_2 = (x_2 - \mu)(x_2 - \mu)^t = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -1.5 & 0 \end{bmatrix} = \begin{bmatrix} 2.25 & 0 \\ 0 & 0 \end{bmatrix}$$

$$m_3 = (x_3 - \mu)(x_3 - \mu)^t = \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} -0.5 & -2 \end{bmatrix} = \begin{bmatrix} 0.25 & 1 \\ 1 & 4 \end{bmatrix}$$

$$m_4 = (x_4 - \mu)(x_4 - \mu)^t = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$m_5 = (x_5 - \mu)(x_5 - \mu)^t = \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 1.5 & 2 \end{bmatrix} = \begin{bmatrix} 2.25 & 3 \\ 3 & 4 \end{bmatrix}$$

$$m_6 = (x_6 - \mu)(x_6 - \mu)^t = \begin{bmatrix} 2.5 \\ 3 \end{bmatrix} \begin{bmatrix} 2.5 & 3 \end{bmatrix} = \begin{bmatrix} 6.25 & 7.5 \\ 7.5 & 9 \end{bmatrix}$$

Now,

Covariance matrix

$= (m_1 + m_2 + m_3 + m_4 + m_5 + m_6) / 6$

On adding the above matrices and dividing by 6, we get-

$$\text{Covariance Matrix} = \frac{1}{6} \begin{bmatrix} 17.5 & 22 \\ 22 & 34 \end{bmatrix}$$

$$\text{Covariance Matrix} = \frac{1}{6}\begin{bmatrix} 17.5 & 22 \\ 22 & 34 \end{bmatrix}$$

$$\text{Covariance Matrix} = \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix}$$

Step-05:

Calculate the eigen values and eigen vectors of the covariance matrix.

λ is an eigen value for a matrix M if it is a solution of the characteristic equation $|M - \lambda I| = 0$.

So, we have-

$$\begin{vmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{vmatrix} - \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} = 0$$

$$\begin{vmatrix} 2.92 - \lambda & 3.67 \\ 3.67 & 5.67 - \lambda \end{vmatrix} = 0$$

From here,

$(2.92 - \lambda)(5.67 - \lambda) - (3.67 \times 3.67) = 0$

$16.56 - 2.92\lambda - 5.67\lambda + \lambda^2 - 13.47 = 0$

$\lambda^2 - 8.59\lambda + 3.09 = 0$

Solving this quadratic equation, we get $\lambda = 8.22, 0.38$

Thus, two eigen values are $\lambda_1 = 8.22$ and $\lambda_2 = 0.38$.

Clearly, the second eigen value is very small compared to the first eigen value.

So, the second eigen vector can be left out.

Eigen vector corresponding to the greatest eigen value is the principal component for the given data set.

So. we find the eigen vector corresponding to eigen value λ_1.

We use the following equation to find the eigen vector-

$$MX = \lambda X$$

where-

- M = Covariance Matrix
- X = Eigen vector
- λ = Eigen value

Substituting the values in the above equation, we get-

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} \begin{bmatrix} X1 \\ X2 \end{bmatrix} = 8.22 \begin{bmatrix} X1 \\ X2 \end{bmatrix}$$

Solving these, we get-

$2.92X_1 + 3.67X_2 = 8.22X_1$

$3.67X_1 + 5.67X_2 = 8.22X_2$

On simplification, we get-

$5.3X_1 = 3.67X_2$ ………(1)

$3.67X_1 = 2.55X_2$ ………(2)

From (1) and (2), $\mathbf{X_1 = 0.69X_2}$

From (2), the eigen vector is-

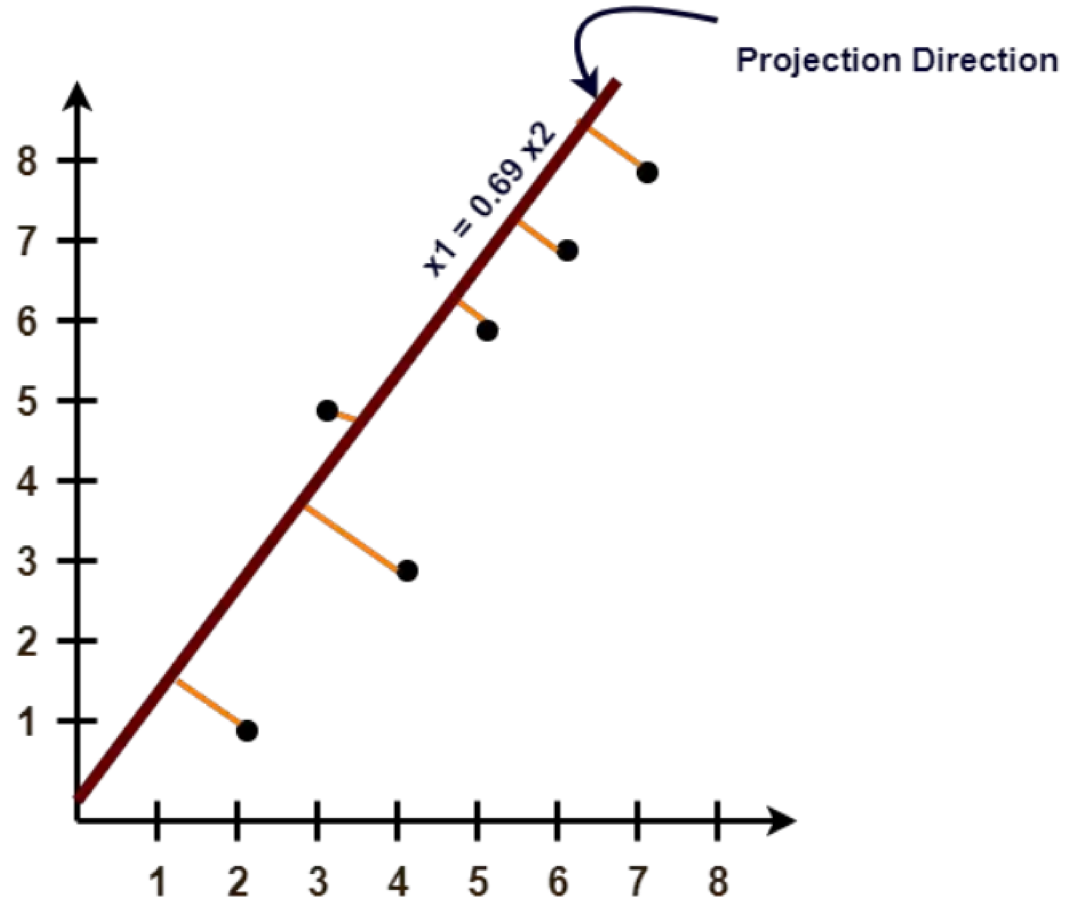
Eigen Vector : $\begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$

From (2), the eigen vector is-

$$\textbf{Eigen Vector :} \begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$

Thus, principal component for the given data set is-

$$\textbf{Principal Component :} \begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$


Projection Direction
x1 = 0.69 x2
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8

The given feature vector is (2, 1).

$$\textbf{Given Feature Vector :} \quad \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

The feature vector gets transformed to

= Transpose of Eigen vector x (Feature Vector – Mean Vector)

$$= \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}^T \times \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 4.5 \\ 5 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 2.55 & 3.67 \end{bmatrix} \times \begin{bmatrix} -2.5 \\ -4 \end{bmatrix}$$

$$= -21.055$$

PCA and standardisation

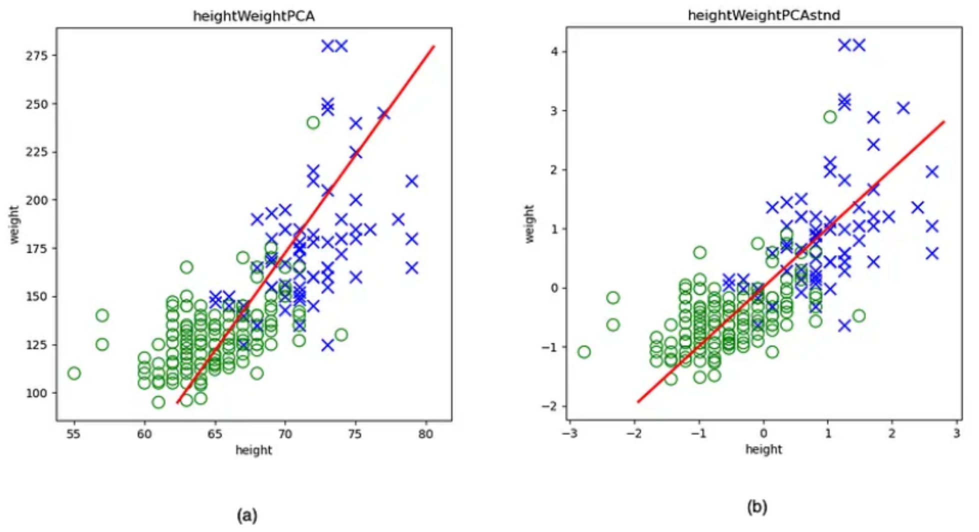


Fig.3 PCA can get mislead by unstandardised data. (a) Principal Component is skewed because PCA is misled by the unstandardised data. (b) PCA when the scales are standardised. Image generated from code at [5].

The principal directions from PCA are the ones along which the variance is the most. So, PCA can be mislead by directions along which the variance appears high just because of the measurement scale. We can see this in Fig. 3(a) where the Principal Component is not aligned properly because it is mislead by the unstandardised scale. Fig. 3(b) shows the correct principal

Example 2:

LARGE SIZE APPLES	ROTTEN APPLES	DAMAGED APPLES	SMALL APPLES
F1	F2	F3	F4
1	5	3	1
4	2	6	3
1	4	3	2
4	4	1	1
5	5	2	3

Calculate the Mean and Standard Deviation for each feature and then, tabulate the same as follows.

	F1	F2	F3	F4
MEAN	3	4	3	2
STANDARD DEVIATION	1.87	1.223	1.87	1

Here in the formula of the standard deviation we used 1/n-1

F1	F2	F3	F4
-1.0695	0.8196	0	-1
0.5347	-1.6393	1.6042	1
-1.0695	0	0	0
0.5347	0	-1.0695	-1
1.0695	0.8196	-0.5347	1

This is the Standardized data set.

Similarly solving all the features, the covariance matrix will be,

	F1	F2	F3	F4
F1	1	-0.218	-0.071	0.534
F2	-0.218	1	-0.763	-0.408
F3	-0.071	-0.763	1	0.534
F4	0.534	-0.408	0.534	1

```
[[ 1.         -0.21821789 -0.07142857  0.53452248]
 [-0.21821789  1.         -0.76376262 -0.40824829]
 [-0.07142857 -0.76376262  1.          0.53452248]
 [ 0.53452248 -0.40824829  0.53452248  1.        ]]
```

Here in the formula for the covariance i used 1/n-1

When you solve the following the matrix by considering 0 on right-hand side, you can define eigen values as

```
Eigenvalues are: [2.26076563 1.18888423 0.08497817 0.46537197]
```

The four eigenvectors are

E1	E2	E3	E4
0.295	-0.561	0.552	0.541
-0.781	-0.267	0.462	-0.323
0.407	0.462	0.664	-0.418
-0.369	0.630	0.197	0.653

```
E-value: [2.26076563 1.18888423 0.08497817 0.46537197]
E-vector [[ 0.29520559 -0.7810274   0.40754004 -0.36981209]
 [-0.56103699 -0.26768116  0.46524277  0.63018525]
 [ 0.55255128  0.4622677   0.66483705  0.19745216]
 [ 0.54108986 -0.32349894 -0.41886989  0.65354282]]
```

And the transformed data…

- If we keep two eigenvectors

2*4 in rows

- And the standardized data is (4 features and 5 records)

4*5 (transposed)

If we multiply we obtain 2*5 (2 principal components and 5 records)

So we reduce the total number of features from 4 into 2.

And the transformed data…

the standardized data is (4 features and 5 records)

5*4

- If we keep two eigenvectors
- 4*2

If we multiply we obtain 5*2 (2 principal components and 5 records)

So we reduce the total number of features from 4 into 2.

And the transformed data (2 eigenvectors are kept)

-1.0695	0.8196	0	-1
0.5347	-1.6393	1.6042	1
-1.0695	0	0	0
0.5347	0	-1.0695	-1
1.0695	0.8196	-0.5347	1

X

E1	E2
0.295	-0.561
-0.781	-0.267
0.407	0.462
-0.369	0.630

5*4

4*2

The space of all face images

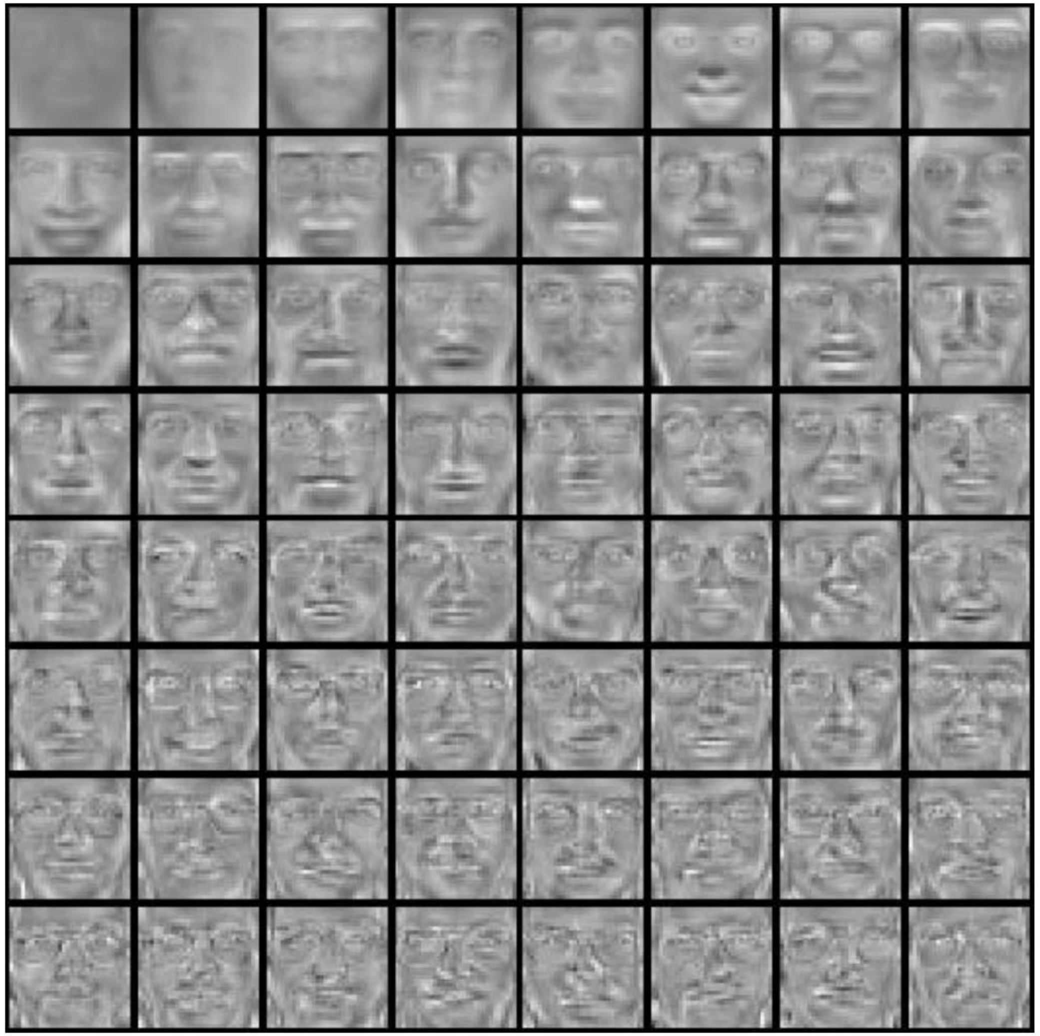
- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
  - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images

slide by Derek Hoiem

Eigenfaces example

Mean: μ

Top eigenvectors: $u_1, \ldots u_k$



slide by Derek Hoiem

Representation and reconstruction

- Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x} - \mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x} - \mu)]$$
$$= \quad w_1, \ldots, w_k$$

- Reconstruction:

 =  + 

$$\hat{x} = \mu + w_1u_1+w_2u_2+w_3u_3+w_4u_4+ \ldots$$

slide by Derek Hoiem

Reconstruction

P = 4

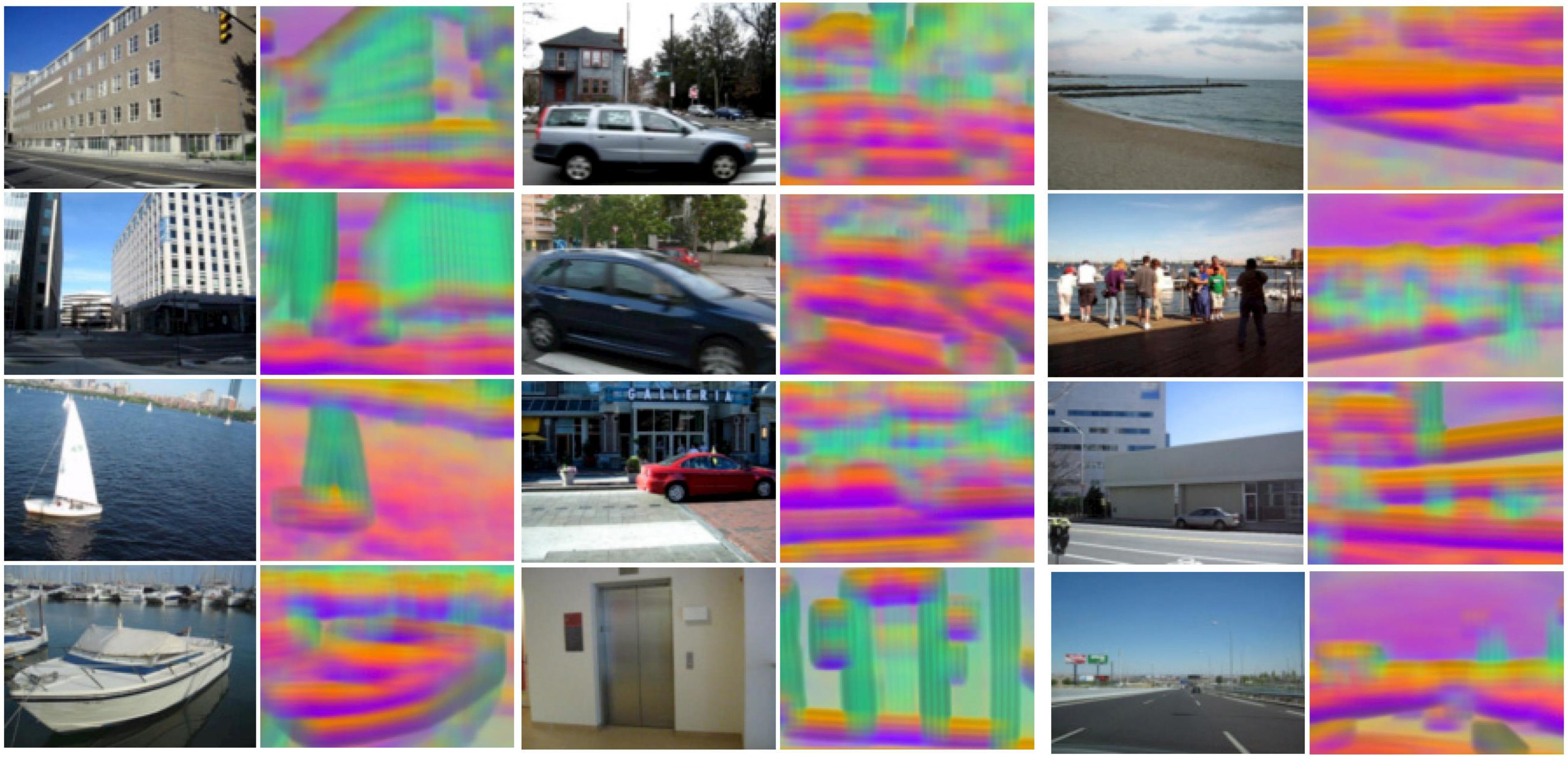
P = 200

P = 400

After computing eigenfaces using 400 face images from ORL face database

slide by Derek Hoiem

SIFT feature visualization



- The top three principal components of SIFT descriptors from a set of images are computed
- Map these principal components to the principal components of the RGB space
- pixels with similar colors share similar structures

Application: Image compression

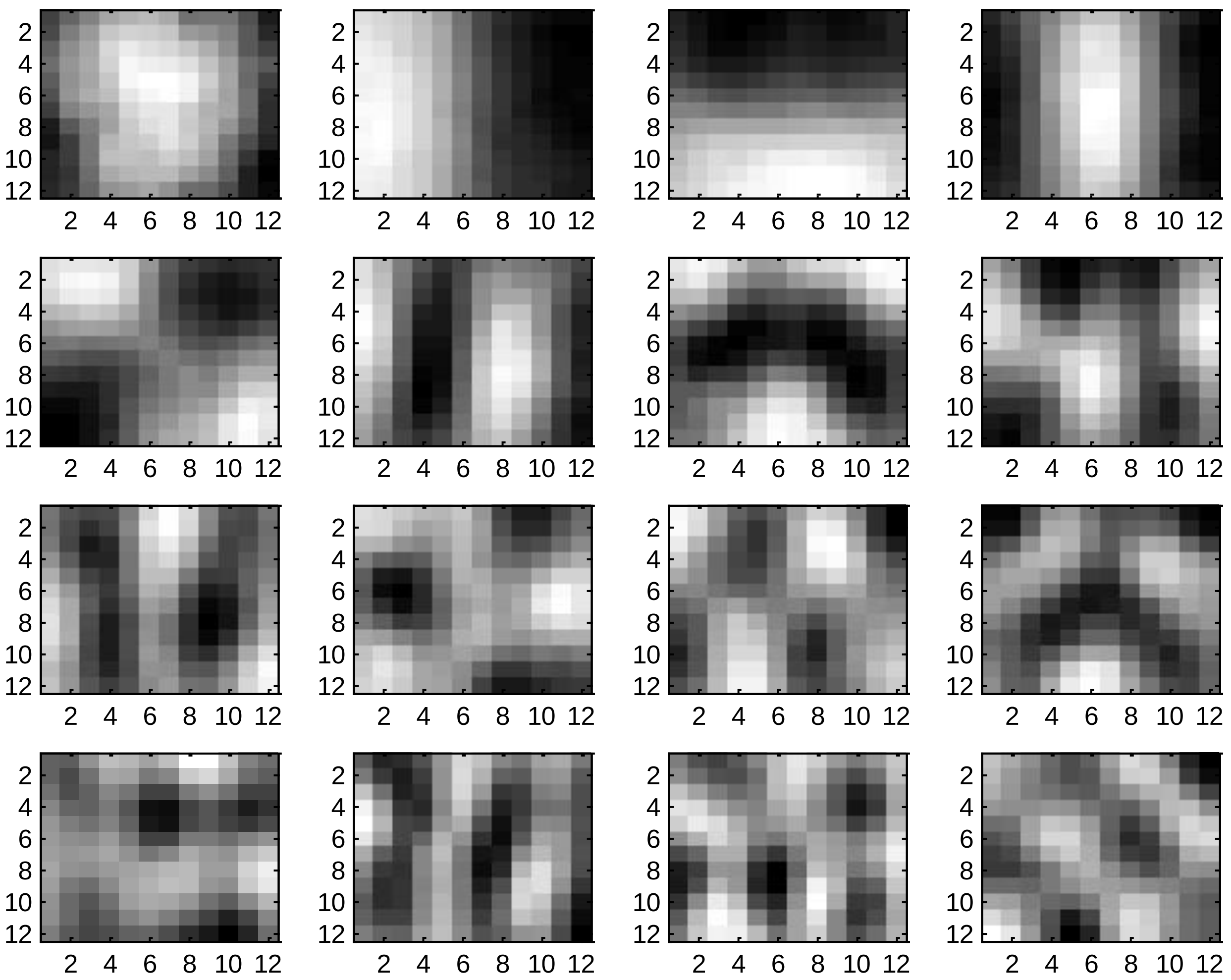
Original Image

- Divide the original 372x492 image into patches:
  - Each patch is an instance that contains 12x12 pixels on a grid
- View each as a 144-D vector

PCA compression: 144D → 60D

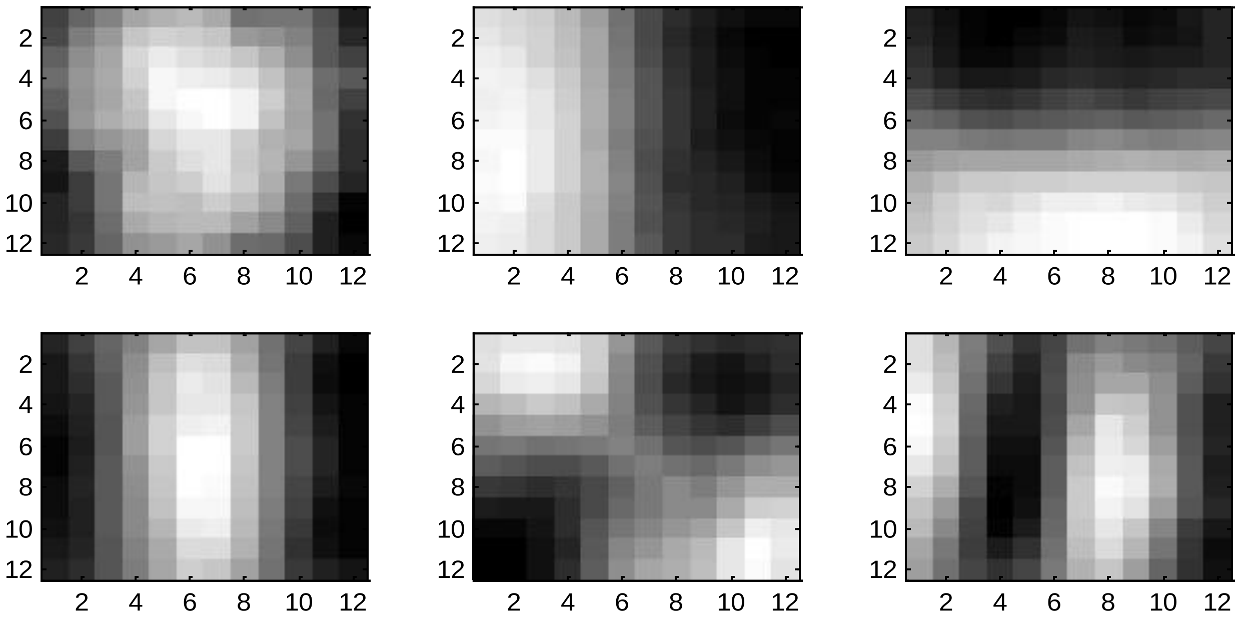
PCA compression: 144D → 16D

16 most important eigenvectors



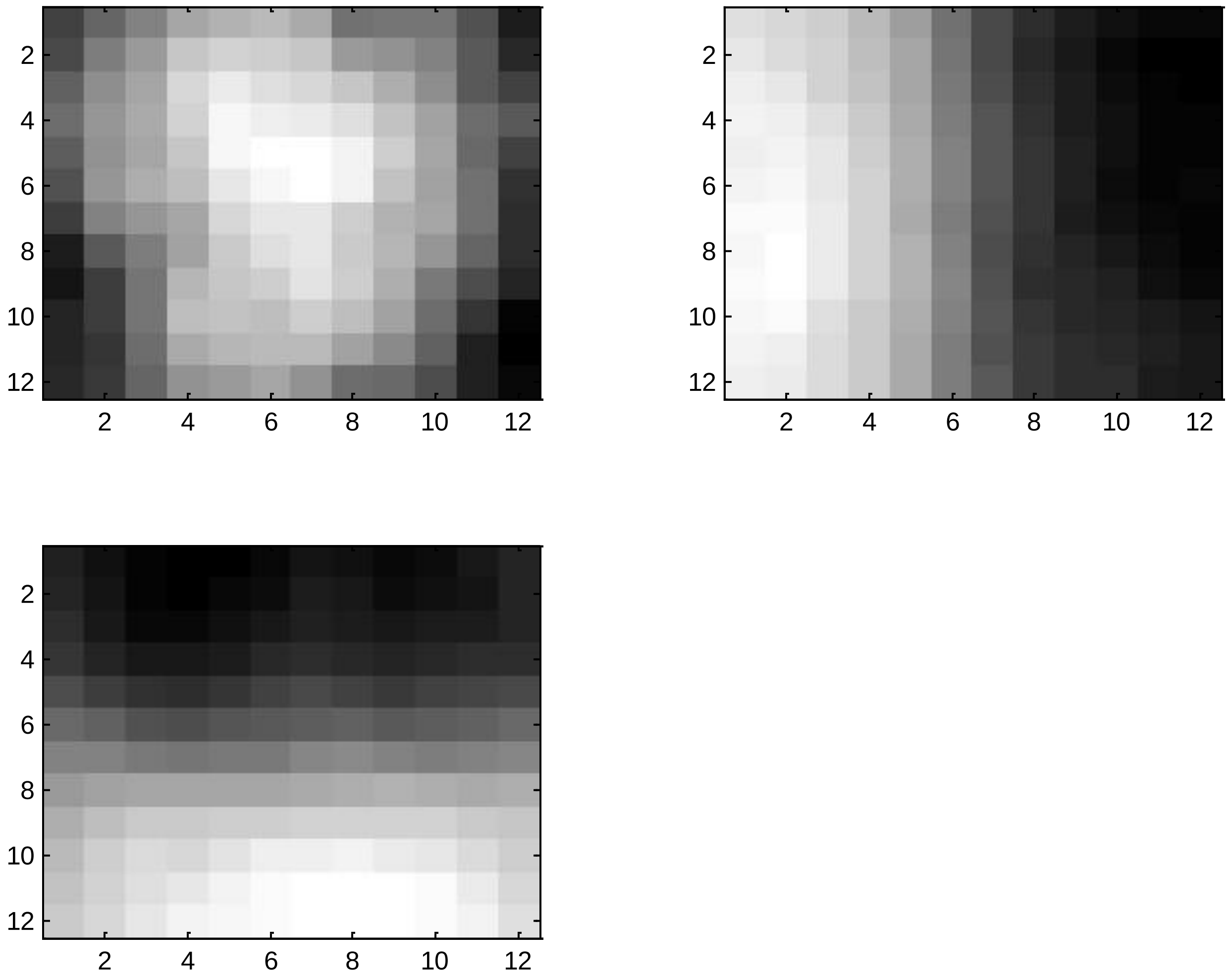
PCA compression: 144D → 6D

6 most important eigenvectors

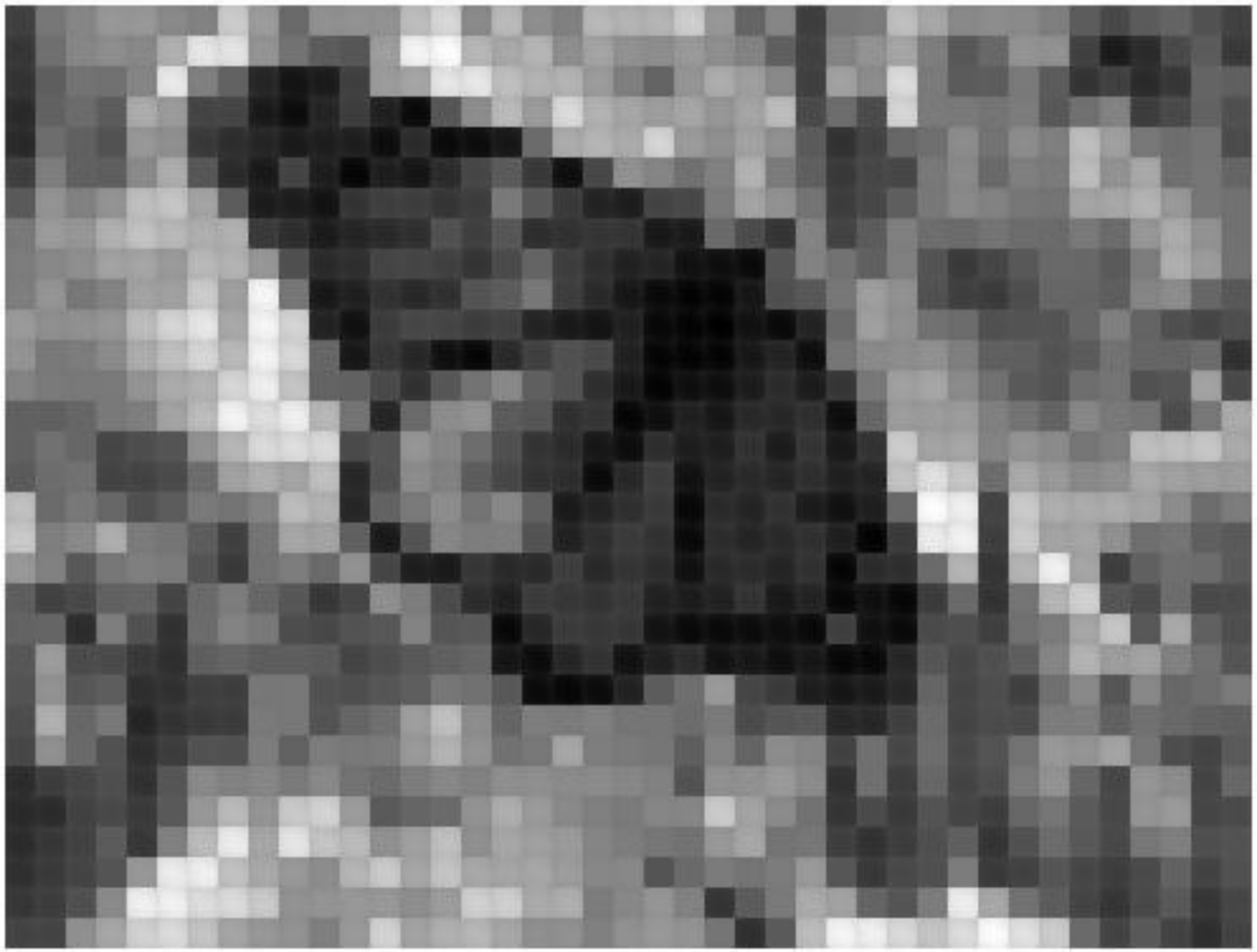


PCA compression: 144D → 3D

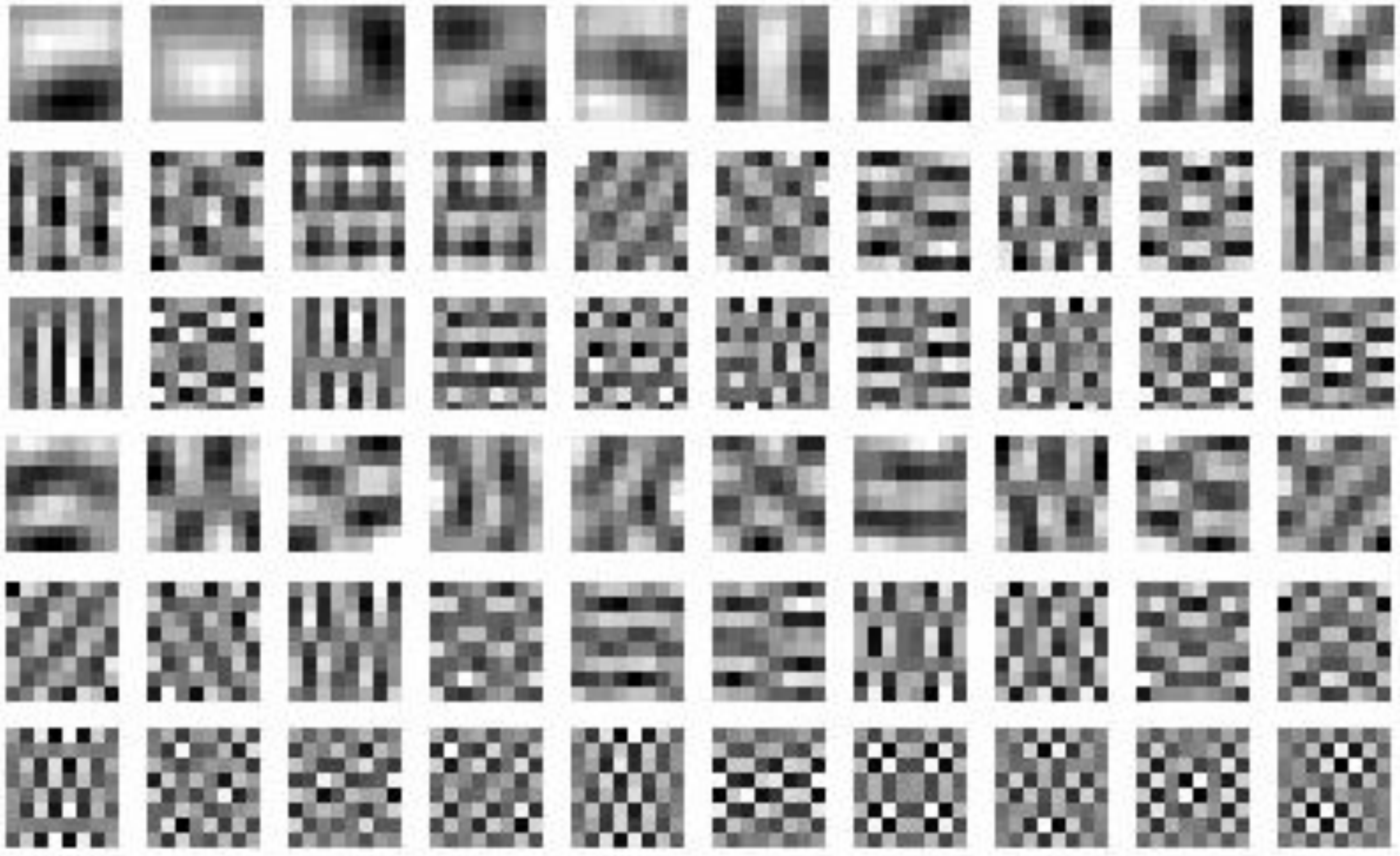
3 most important eigenvectors



PCA compression: 144D → 1D

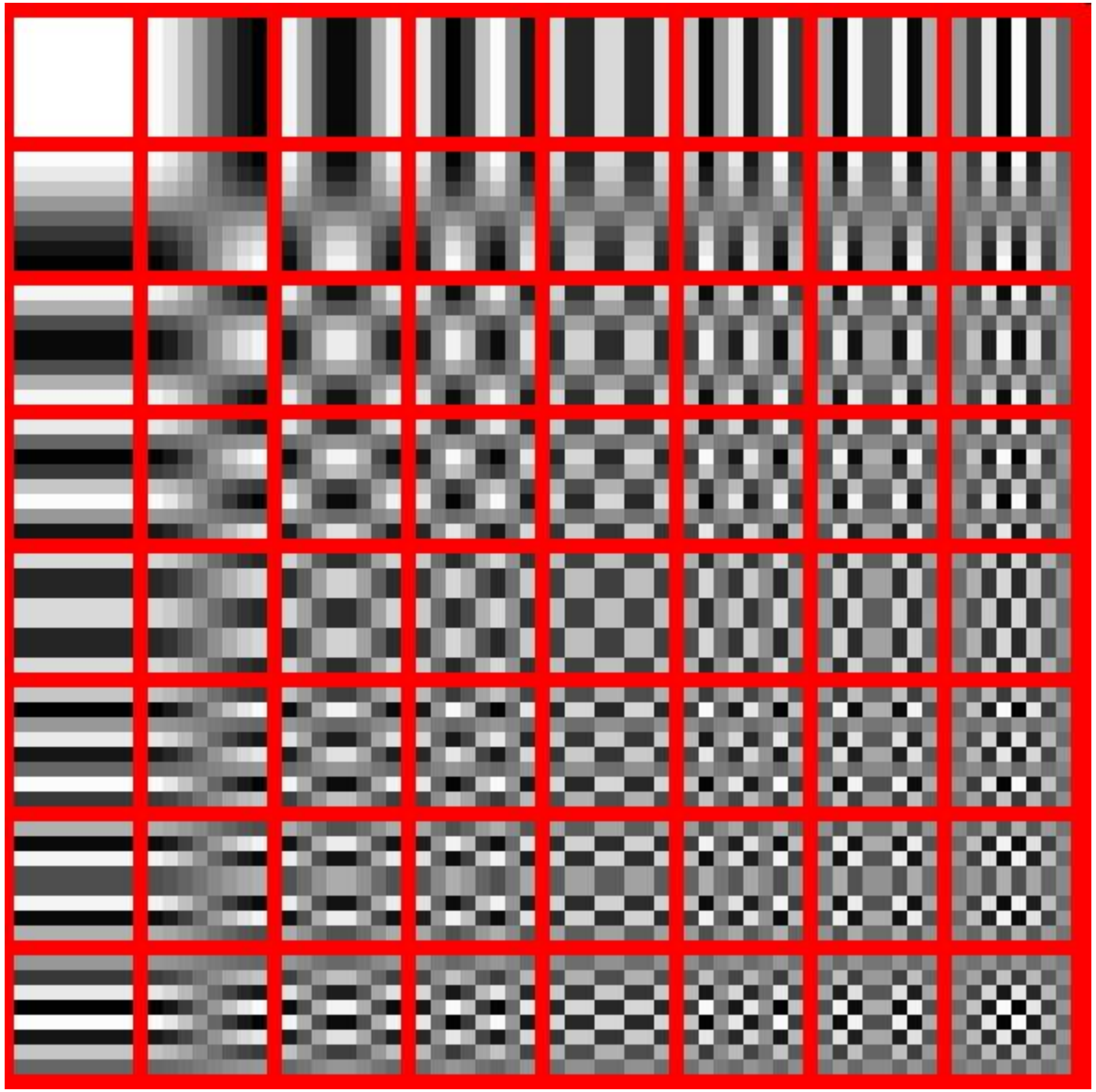


60 most important eigenvectors



Looks like the discrete cosine bases of JPG!...

2D Discrete Cosine Basis



http://en.wikipedia.org/wiki/Discrete_cosine_transform