Machine Learning

Principal Component Analysis (PCA)

Digital Vidya

Principal Components

- General about principal components
  - summary variables
  - linear combinations of the original variables
  - uncorrelated with each other
  - capture as much of the original variance as possible

Principal Components Analysis (PCA)

- An exploratory technique used to reduce the dimensionality of the data set to 2D or 3D

- Can be used to:
  - Reduce number of dimensions in data
  - Find patterns in high-dimensional data
  - Visualize data of high dimensionality

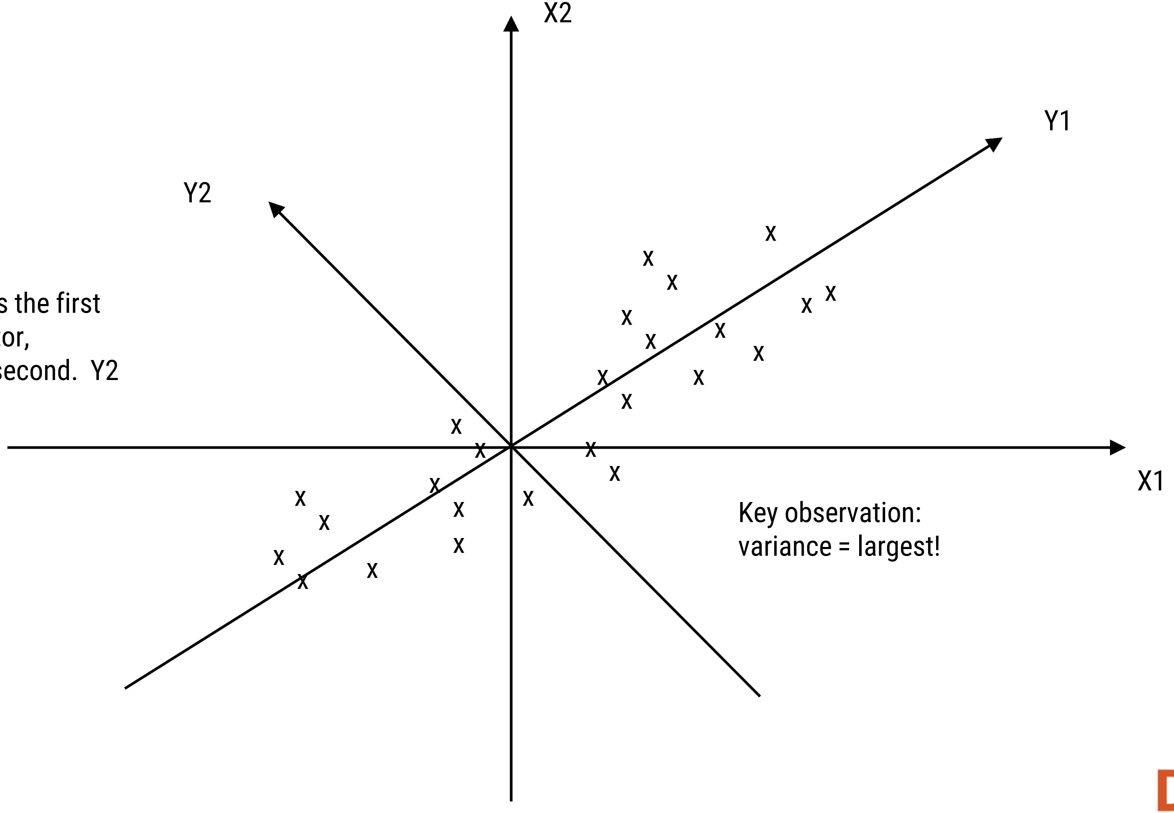
- Example applications:
  - Face recognition
  - Image compression
  - Gene expression analysis

Digital Vidya

Principal Components Analysis Ideas (PCA)

- Does the data set ‘span’ the whole of d dimensional space?
- For a matrix of *m* samples x *n* genes, create a new covariance matrix of size *n* x *n*.
- Transform some large number of variables into a smaller number of uncorrelated variables called principal components (PCs).
- Developed to capture as much of the variation in data as possible

Principal Component Analysis



Note: Y1 is the first eigen vector,
Y2 is the second. Y2 ignorable.

Key observation:
variance = largest!

Principal Component Analysis: 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

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

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

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

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):

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

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 $A\mathbf{x}=\lambda\mathbf{x}$, λ is called an *eigenvalue* of *A*.

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

Eigenvalues & eigenvectors

- $A\mathbf{x}=\lambda\mathbf{x} \Leftrightarrow (A-\lambda \mathrm{I})\mathbf{x}=0$

- How to calculate $\mathbf{x}$ 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 $\mathbf{x}$

Principal components

- 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 greatest variation
- Principal component (PC2)
  - the direction with maximum variation left in data, orthogonal to PC1
- In general, only few directions manage to capture most of the variability in the data.

Steps of PCA

- Let $\overline{X}$ be the mean vector (taking the mean of all rows)
- Adjust the original data by the mean

  $X' = X - \overline{X}$
- Compute the covariance matrix C of adjusted X
- Find the eigenvectors and eigenvalues of C.

- For matrix *C*, vectors **e** (=column vector) having same direction as *C***e** :
  - *eigenvectors* of *C* is **e** such that *C***e**=λ**e**,
  - λ is called an *eigenvalue* of *C*.
- *C***e**=λ**e** $\Leftrightarrow$ (*C*-λI)**e**=0
  - **Most data mining packages do this for you.**

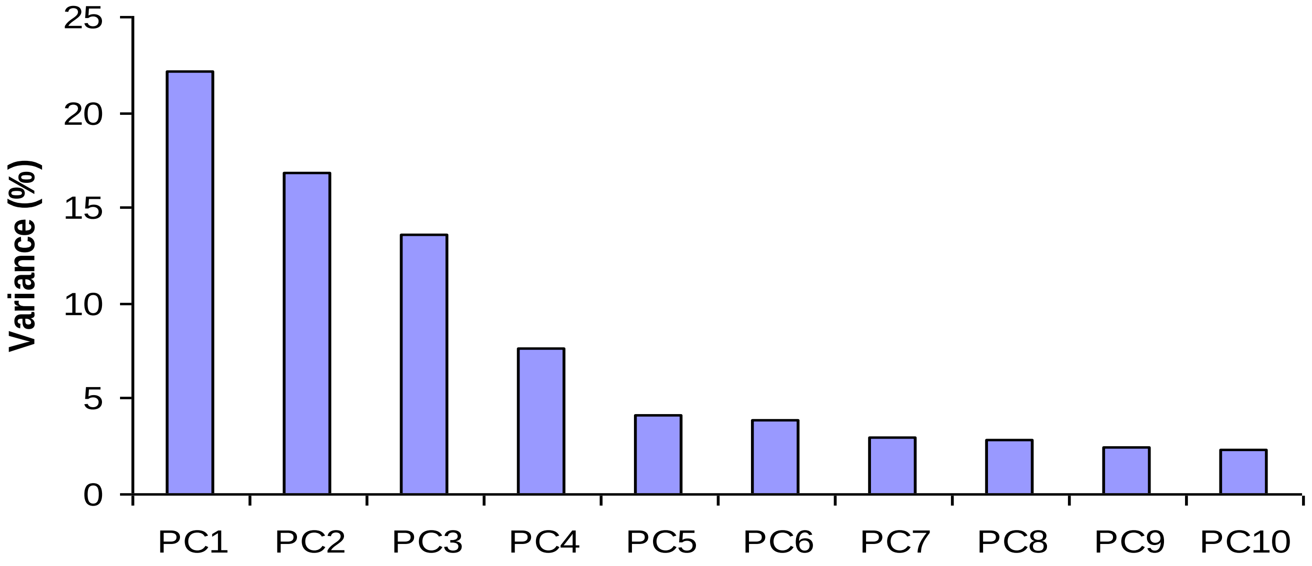
Digital Vidya

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:

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

Principal components - Variance



Transformed Data

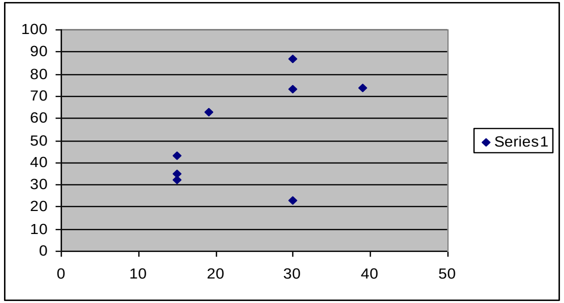
- Eigenvalues λ_j corresponds to variance on each component *j*
- *Thus, sort by* λ_j
- Take the first *p* eigenvectors $\mathbf{e}_i$; where p is the number of top eigenvalues
- These are the directions with the largest variances

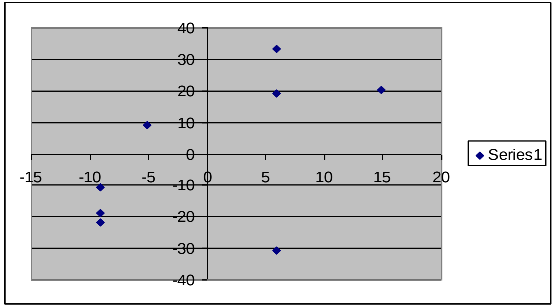
$$\begin{pmatrix} y_{i1} \\ y_{i2} \\ \cdots \\ y_{ip} \end{pmatrix} = \begin{pmatrix} e_1 \\ e_2 \\ \cdots \\ e_p \end{pmatrix} \begin{pmatrix} x_{i1} - \overline{x_1} \\ x_{i2} - \overline{x_2} \\ \cdots \\ x_{in} - \overline{x_n} \end{pmatrix}$$

Example 1

Mean1=24.1
Mean2=53.8

X1	X2	X1'	X2'
19	63	-5.1	9.25
39	74	14.9	20.25
30	87	5.9	33.25
30	23	5.9	-30.75
15	35	-9.1	-18.75
15	43	-9.1	-10.75
15	32	-9.1	-21.75
30	73	5.9	19.25





Covariance Matrix

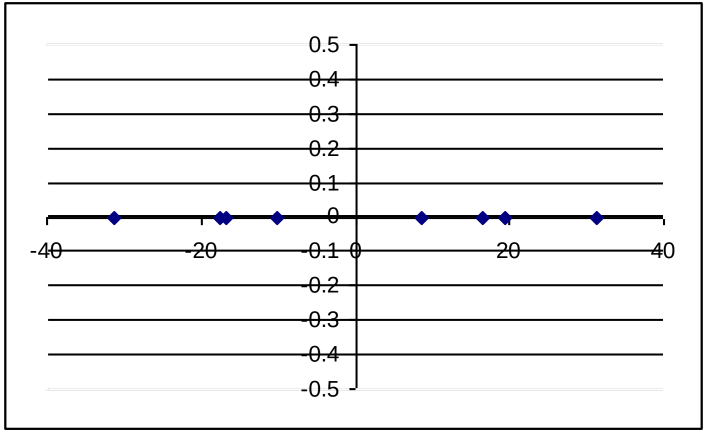
- C =

75	106
106	482

- Using Algebraic rules, we find out:
  - Eigenvectors:
  - e1=(-0.98,-0.21), $\lambda 1$=51.8
  - e2=(0.21,-0.98), $\lambda 2$=560.2
  - Thus the second eigenvector is more important!

If we only keep one dimension: e2

- We keep the dimension of e2=(0.21,-0.98)
- We can obtain the final data as



yi
-10.14
-16.72
-31.35
31.374
16.464
8.624
19.404
-17.63

$$y_i = \begin{pmatrix} 0.21 & -0.98 \end{pmatrix} \begin{pmatrix} x_{i1} \\ x_{i2} \end{pmatrix} = 0.21 * x_{i1} - 0.98 * x_{i2}$$

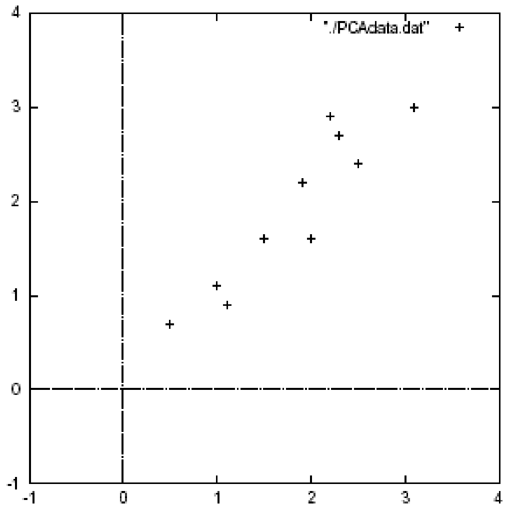
Example 2

Data =

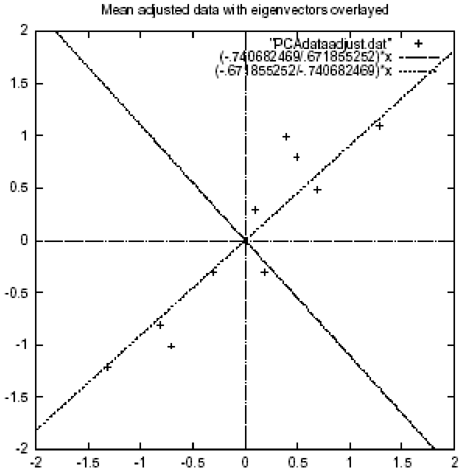
x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01



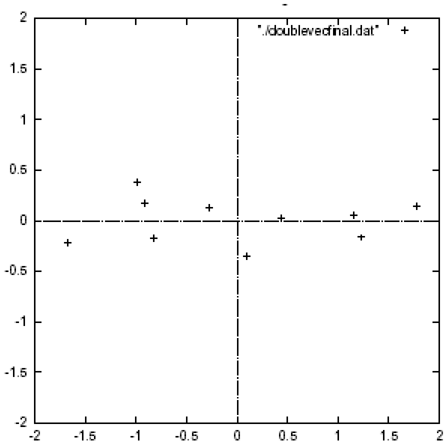
Example 2



Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

Data transformed with 2 eigenvectors



Digital Vidya

PCA –> Original Data

- Retrieving old data (e.g. in data compression)
  - *RetrievedRowData=(RowFeatureVector*^{T} x *FinalData)+OriginalMean*
  - Yields original data using the chosen components

Thank You