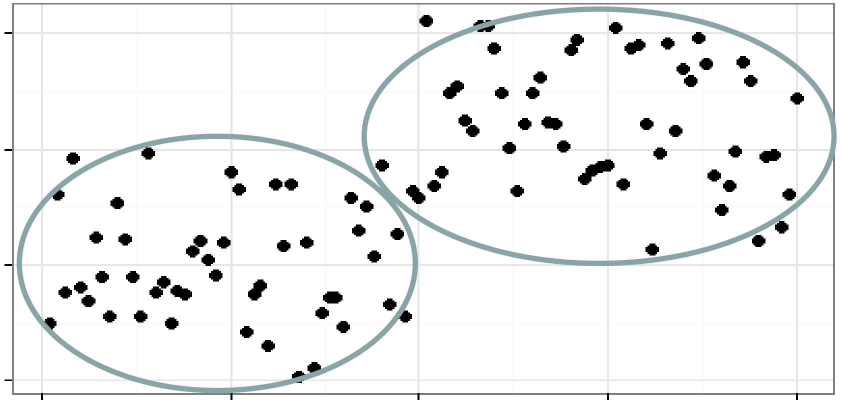
Inspire...Educate...Transform.

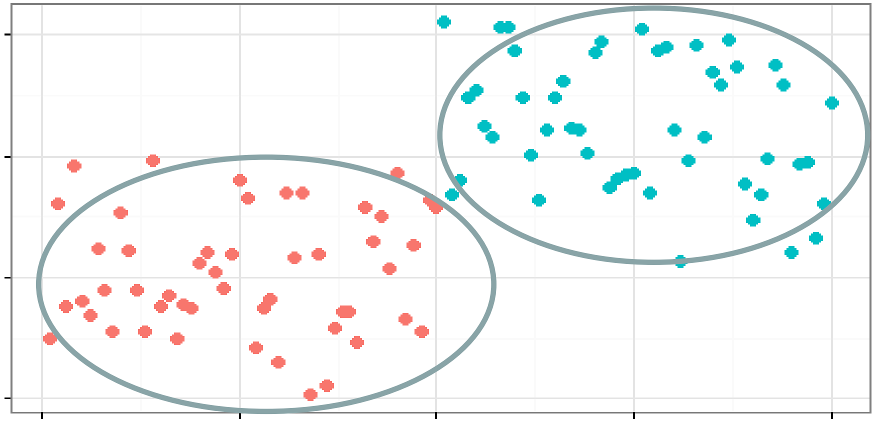
Inspire…Educate…Transform.

Clustering and Linear algebra

Dr. Kishore Reddy Konda

Mentor, INSOFE





Clustering

- Finding similarity groups in data, called **clusters**. I.e.,
  - Data instances that are similar to (near) each other are in the same cluster
  - Data instances that are very different (far away) from each other fall in different clusters.

A Few Clustering Applications

- In marketing, segment customers according to their similarities
- It is not uncommon to have over 100,000 segments in insurance clustering
- Given a collection of text documents, organize them according to their content similarities,
  - E.g., Google news
- Blind signal separation (separating two speakers)

Algorithms

- **<u>Hierarchical</u>** <u>approach</u>: Create a hierarchical decomposition of the set of data (or objects) using some criterion (Wald)

- **<u>Partitioning</u>** <u>approach</u>: Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors (K-means, Spectral clustering)

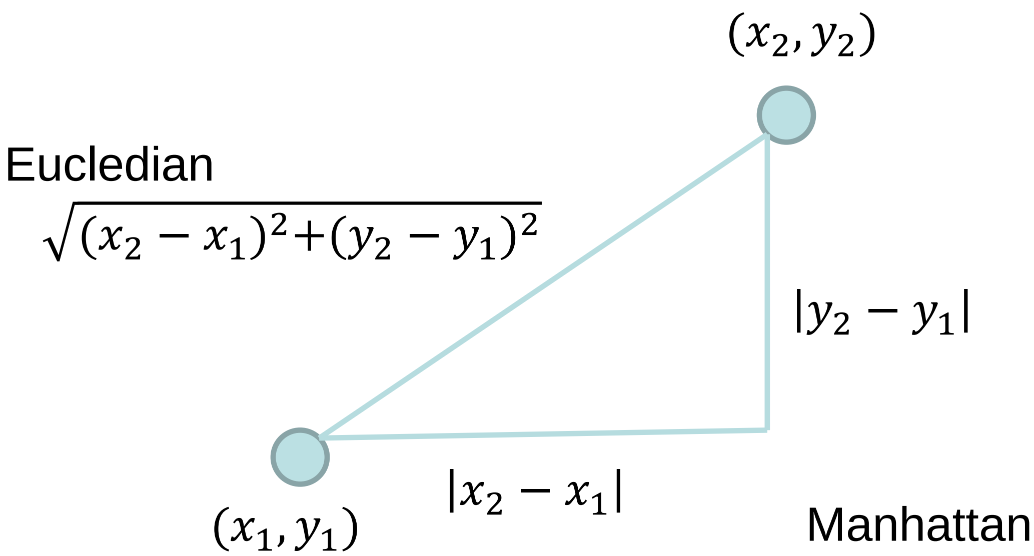
- **<u>Model-based</u>** <u>methods:</u> A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other (EM)

UNDERSTANDING DISTANCE

Desiderata for proximity

- If d_1 is near d_2, then d_2 is near d_1.
- If d_1 near d_2, and d_2 near d_3, then d_1 is not far from d_3.
- No document is closer to d than d itself.

Numeric



Some Other Common Metrics

- Weighted distances
  - More important variables get higher weights

- Minkowski distance
  - $\sqrt[n]{(x_2 - x_1)^n + (y_2 - y_1)^n}$

- The maximum distance amongst all attributes

- Correlation between rows

Norms

$$The\ inner\ product\ generated\ norm\ \|X\|_W = \sqrt{< X, \bar{X} >_W}$$

$$Eucledean\ or\ l_2 norm\ \|X\|_2 = \sqrt{X.\bar{X}}$$

$$The\ l_1 norm\ \|X\|_1 = |x_1| + |x_2| + |x_3| + \cdots + |x_n|$$

$$The\ l_\infty norm\ \|X\|_\infty = \max(|x_1|, |x_2|, |x_3|, \ldots |x_n|)$$

$$The\ l_p norm\ \|X\|_p = (|x_1|^p + |x_2|^p + |x_3|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

Frobenius norm of a matrix is Eucledean version

$$\|A\|_F = \sqrt{\sum \sum |a|_{ij}{}^2}$$

Similarity Measure for documents

- We now have vectors for all documents in the collection, a vector for the query, how to compute similarity?

- Using a similarity measure between the query and each document:

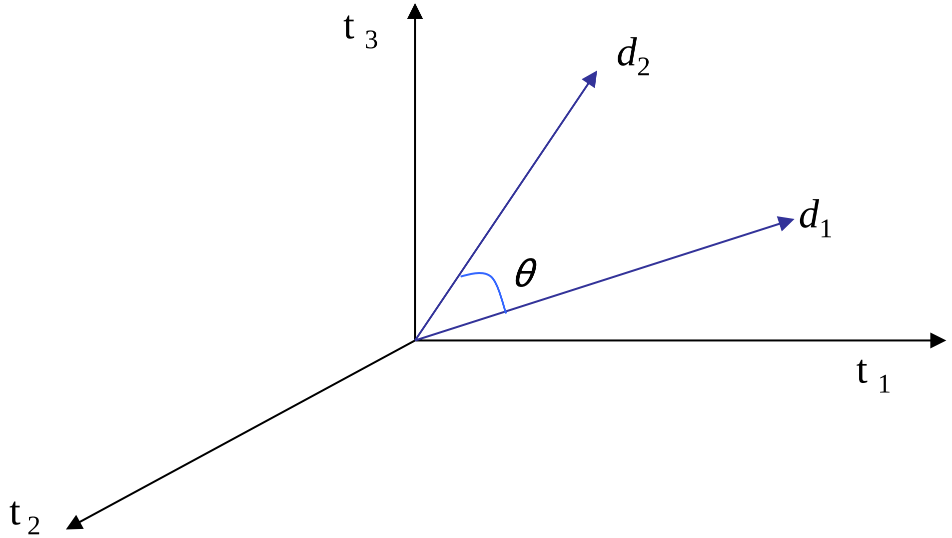
  – It is possible to rank the retrieved documents in the order of presumed relevance (query-dependent ranking).

  – It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

Cosine similarity

- Distance between vectors d_1 and d_2 *captured* by the cosine of the angle x between them.

- Note – this is actually *similarity*, not distance



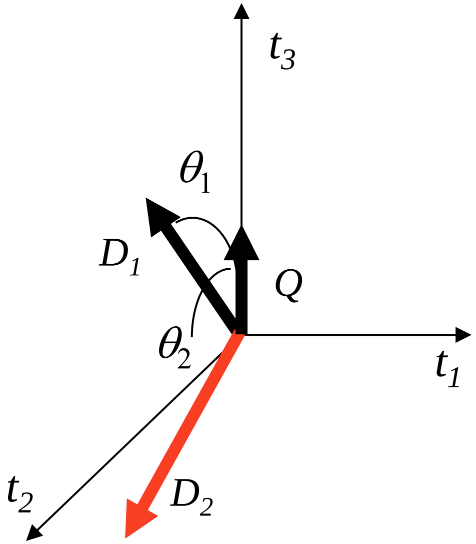
Cosine similarity

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

- Cosine of angle between two vectors
- The denominator involves the lengths of the vectors
- The cosine measure is also known as the *normalized inner product*

Cosine Similarity vs. Inner Product

$D_1 = 2T_1 + 3T_2 + 5T_3$ $\text{CosSim}(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$
$D_2 = 3T_1 + 7T_2 + 1T_3$ $\text{CosSim}(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$
$Q = 0T_1 + 0T_2 + 2T_3$



D_1 is 6 times better than D_2 using cosine similarity but only 5 times better using inner product.

Categorical Attributes in Unsupervised Settings

- Unsupervised setting
  - Approach 1: Create dummies and use the same metric you use for numeric attributes

Attribute
1
2
3

→

Attribute	a1	a2	a3
1	1	0	0
2	0	1	0
3	0	0	1

Categorical Attributes in Unsupervised Settings: II

		Data point j		
		1	0	
Data point i	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	$a+b+c+d$

$$Hamming\ distance = \frac{\#of\ dissimilar\ attributes}{\#of\ dissimilar + \#of\ similar} = \frac{b+c}{b+c+a+d}$$

Asymmetric Binary Attributes

- Asymmetric: if one of the states is more important or more valuable than the other.
  - By convention, state 1 represents the more important state, which is typically the rare or infrequent state.
  - Jaccard coefficient is a popular measure
  - We can have some variations, adding weights

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b+c}{a+b+c}$$

Dissimilarity Between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - Gender is a symmetric attribute
  - The remaining attributes are asymmetric binary
  - Let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Cat variables: Supervised learning

Cat Attribute	C1	C2	C3
L1	78	22	0
L2	90	7	3
L3	77	22	1
L4			
L5			
L6			

Use this information for clustering or distances
R1: L1, R2: L2, R3:L3
R1-R2 is farther than R1-R3:

Ordinal Variables

- Same as numeric
- Look up is better than computation

Look Up Matrix for Ordinal with 3 States

$$
\begin{array}{cc}
 & \begin{matrix} 1 & 2 & 3 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{bmatrix}
\end{array}
$$

BACK TO MODELS

HIERARCHICAL (AGGLOMERATIVE) CLUSTERING

Example of Agglomerative Clustering

	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NY	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

	BOS/NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY	0	223	1308	802	2815	2934	2786	1771
DC	223	0	1075	671	2684	2799	2631	1616
MIA	1308	1075	0	1329	3273	3053	2687	2037
CHI	802	671	1329	0	2013	2142	2054	996
SEA	2815	2684	3273	2013	0	808	1131	1307
SF	2934	2799	3053	2142	808	0	379	1235
LA	2786	2631	2687	2054	1131	379	0	1059
DEN	1771	1616	2037	996	1307	1235	1059	0

	BOS/NY/DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY/DC	0	1075	671	2684	2799	2631	1616
MIA	1075	0	1329	3273	3053	2687	2037
CHI	671	1329	0	2013	2142	2054	996
SEA	2684	3273	2013	0	808	1131	1307
SF	2799	3053	2142	808	0	379	1235
LA	2631	2687	2054	1131	379	0	1059
DEN	1616	2037	996	1307	1235	1059	0

	BOS/ NY/DC	MIA	CHI	SEA	SF/LA	DEN
BOS/NY/DC	0	1075	671	2684	2631	1616
MIA	1075	0	1329	3273	2687	2037
CHI	671	1329	0	2013	2054	996
SEA	2684	3273	2013	0	808	1307
SF/LA	2631	2687	2054	808	0	1059
DEN	1616	2037	996	1307	1059	0

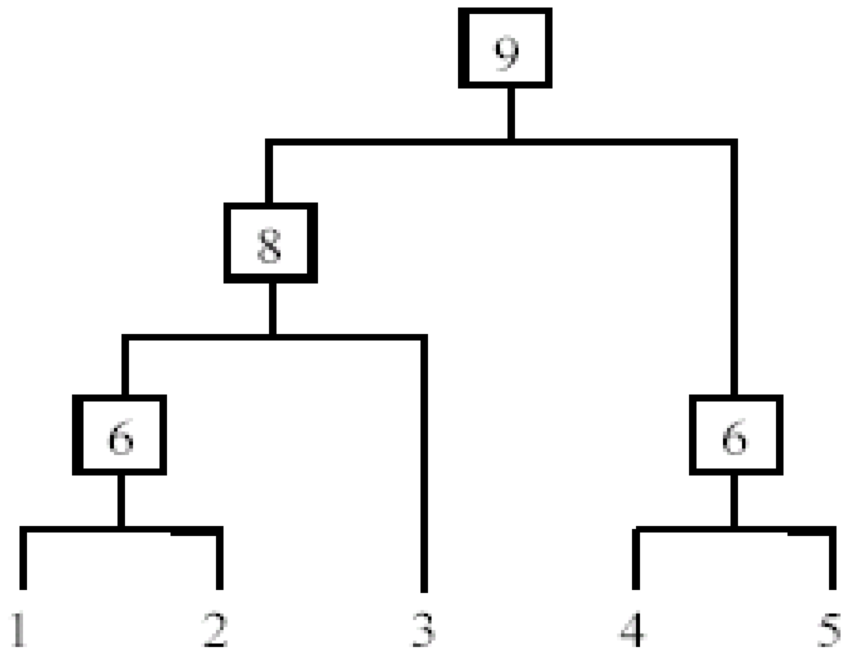
	BOS/NY/DC/ CHI	MIA	SEA	SF/LA	DEN
BOS/NY/DC/CHI	0	1075	2013	2054	996
MIA	1075	0	3273	2687	2037
SEA	2013	3273	0	808	1307
SF/LA	2054	2687	808	0	1059
DEN	996	2037	1307	1059	0

	BOS/NY/DC/CHI	MIA	SF/LA/SEA	DEN
BOS/NY/DC/CHI	0	1075	2013	996
MIA	1075	0	2687	2037
SF/LA/SEA	2054	2687	0	1059
DEN	996	2037	1059	0

	BOS/NY /DC/CHI/DEN	MIA	SF/LA/SEA
BOS/NY/DC/CHI/DEN	0	1075	1059
MIA	1075	0	2687
SF/LA/SEA	1059	2687	0

	BOS/NY /DC/CHI /DEN/SF /LA/SEA	MIA
BOS/NY/DC/CHI/DEN/SF/LA/SEA	0	1075
MIA	1075	0

Hierarchical Clustering



Decomposes data objects into several levels of nested partitioning (tree of clusters).

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

Agglomerative Clustering (Hierarchical)

- Assign each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.
- Merge most similar clusters into a single cluster, so that now you have one less cluster.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

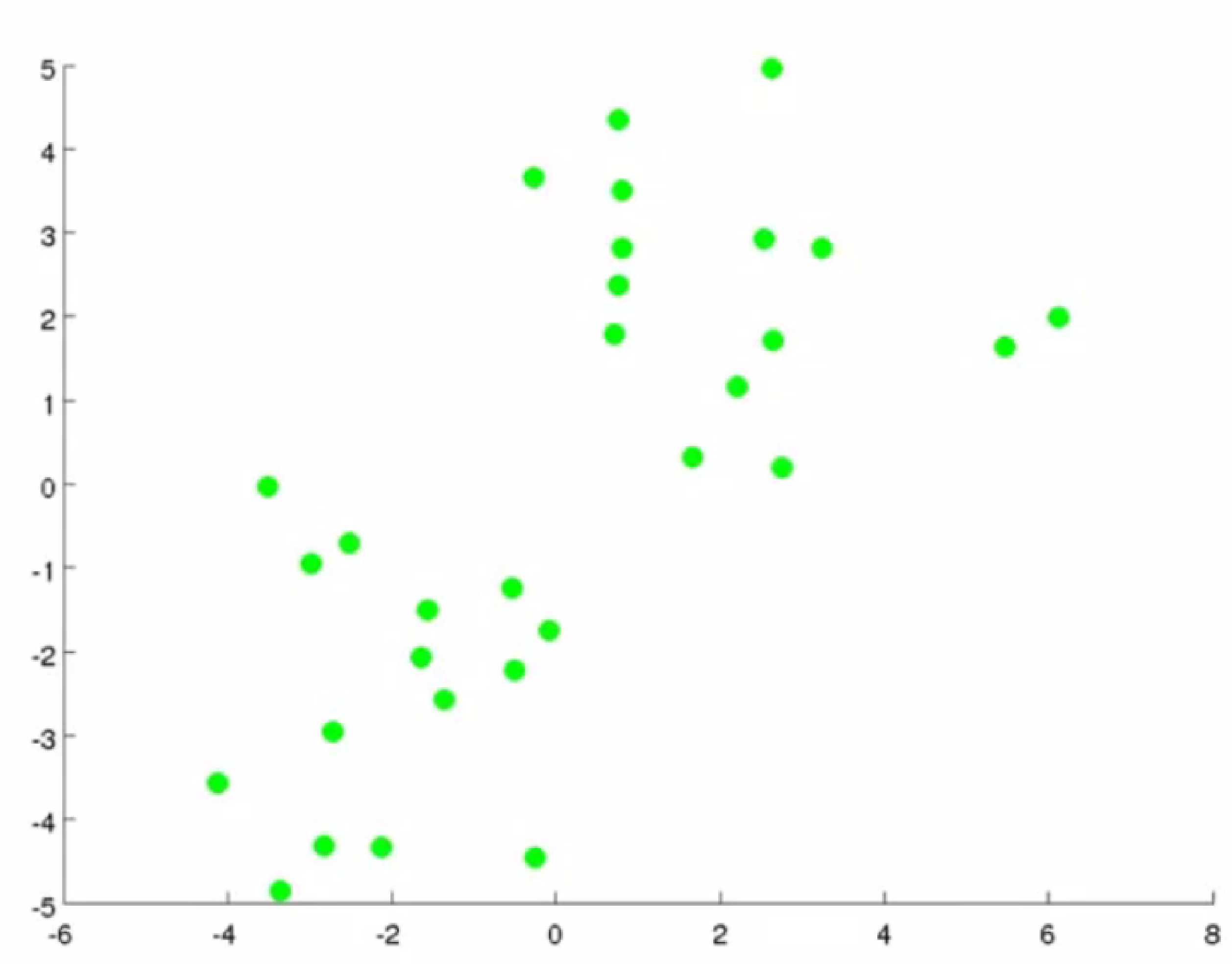
Pvclust: Stability Experiment

- A number of bootstrapped samples
- See how many times a cluster is formed at that level
- Declare a probability value for the cluster based on the count

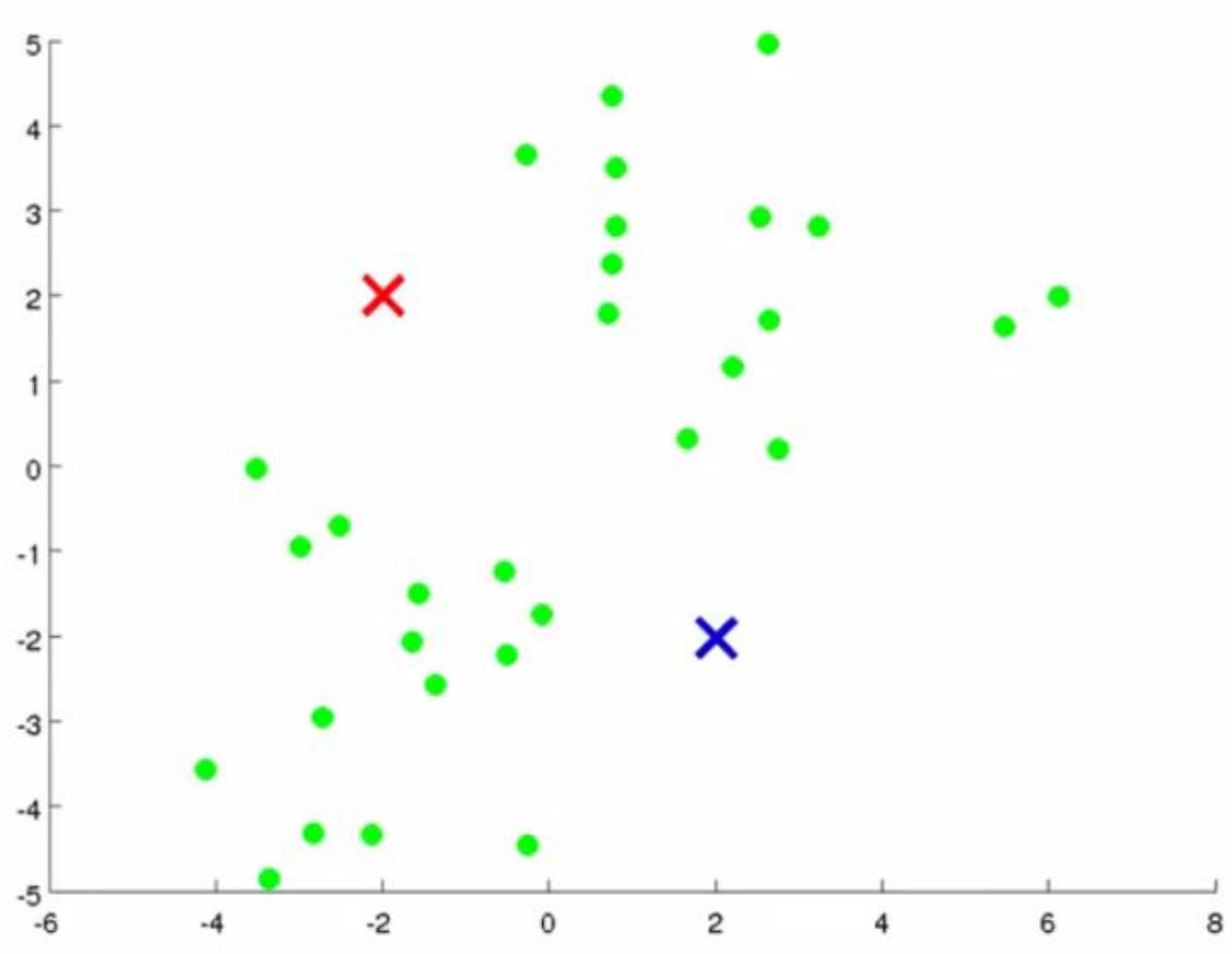
PARTITIONING ALGORITHMS: K-MEANS & K-MEDOIDS

K-Means Clustering

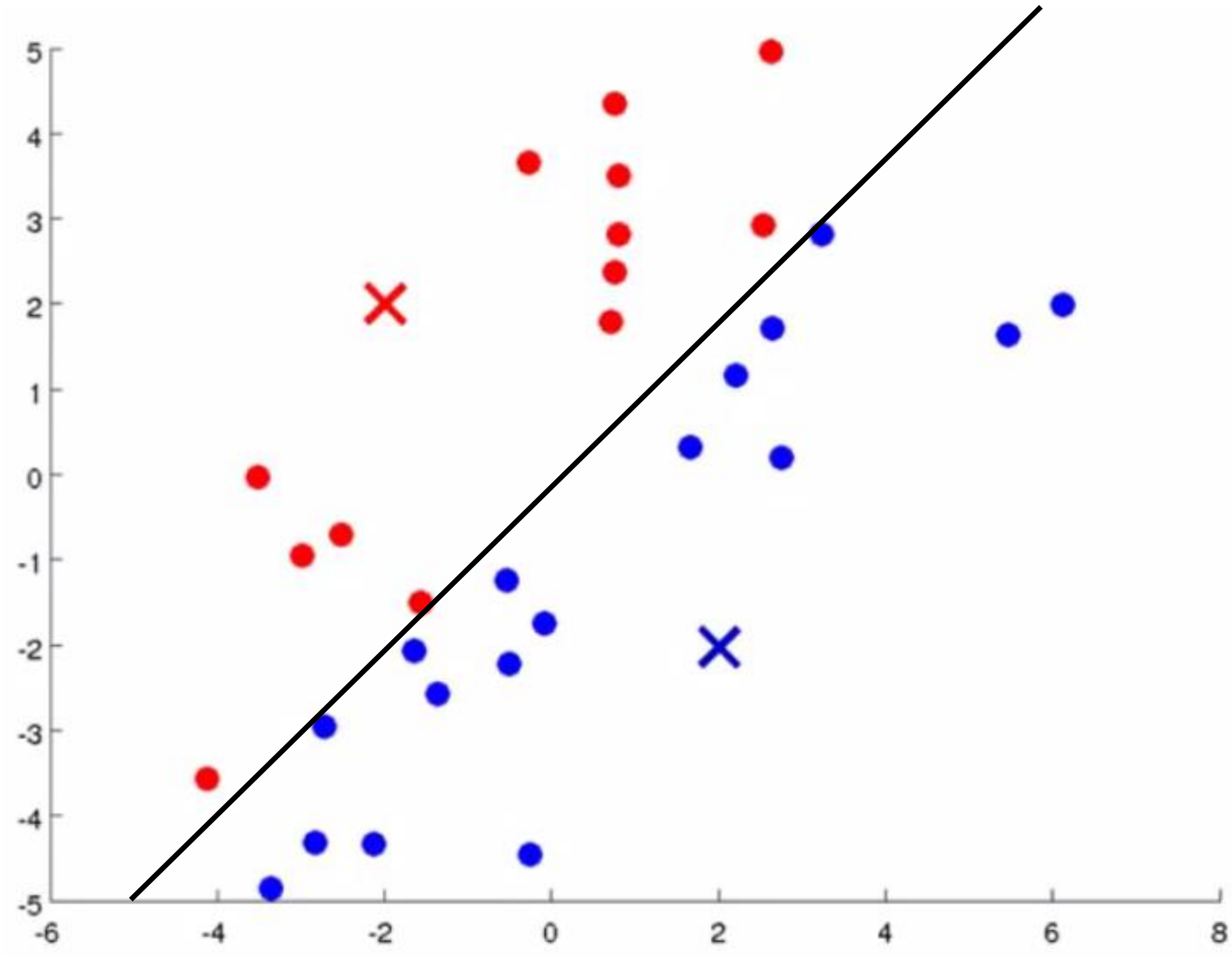
- K-means is a partitional clustering algorithm as it partitions the given data into k clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user



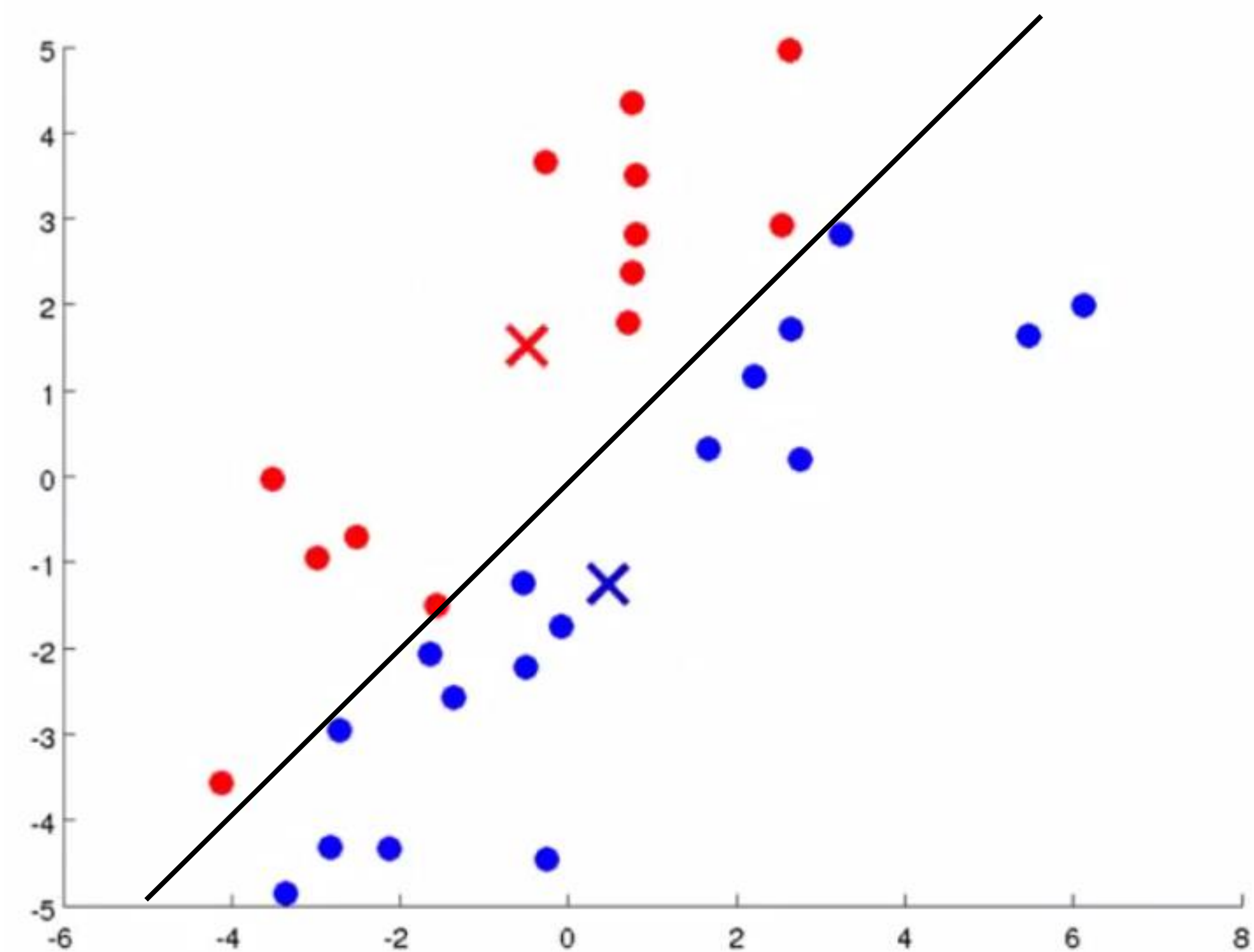
Pictures courtesy of Andrew Ng's course on Coursera.

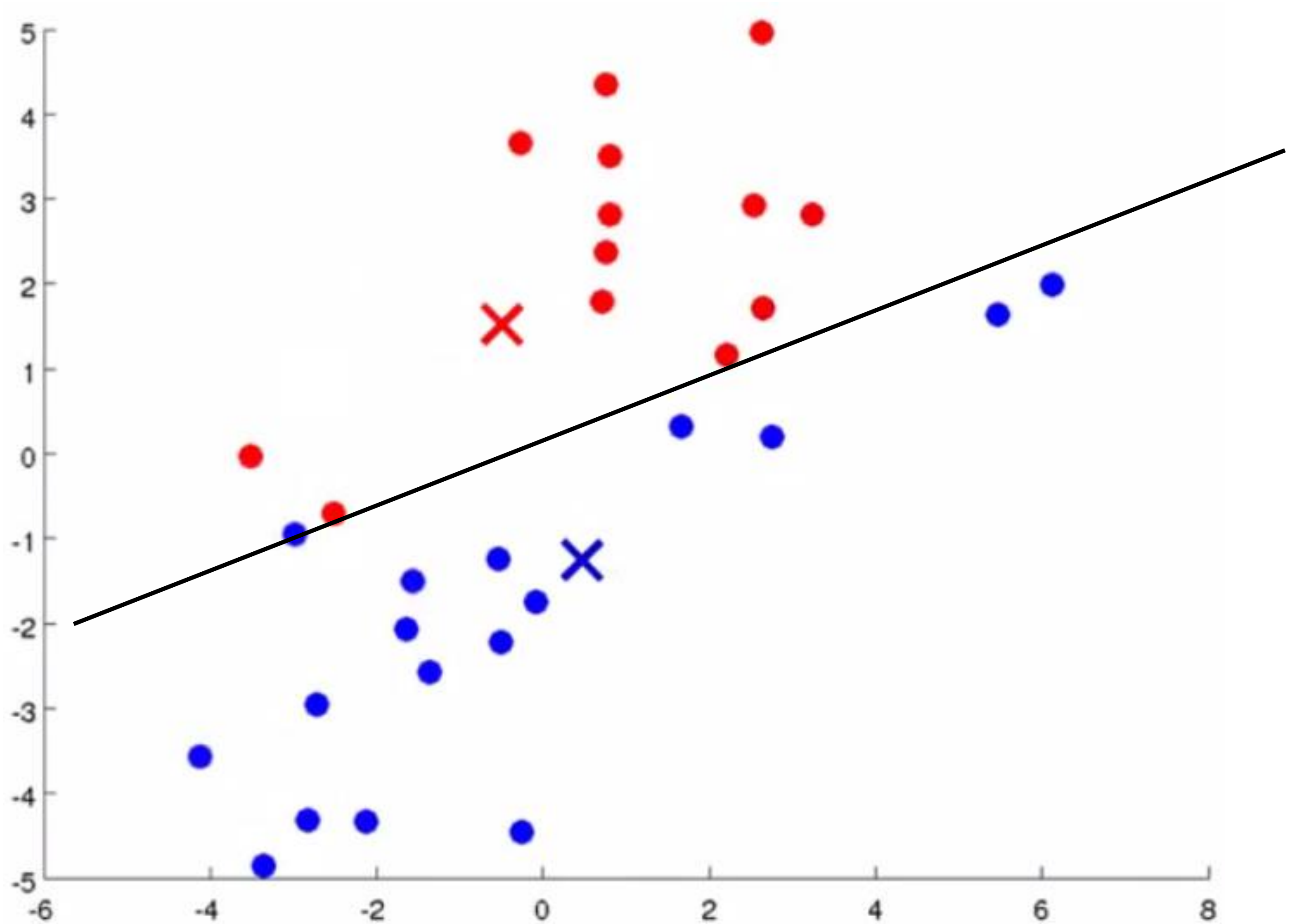


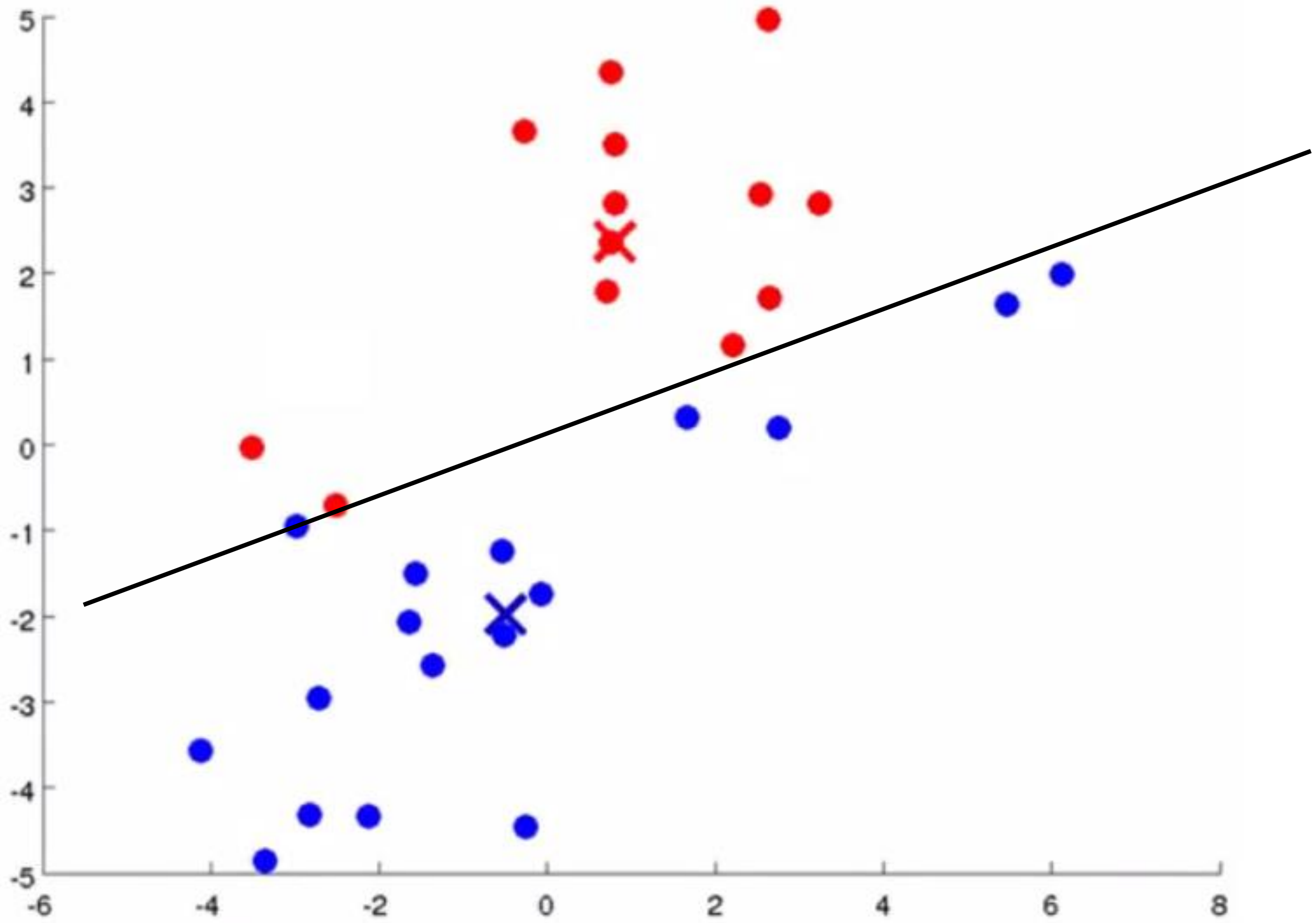
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8



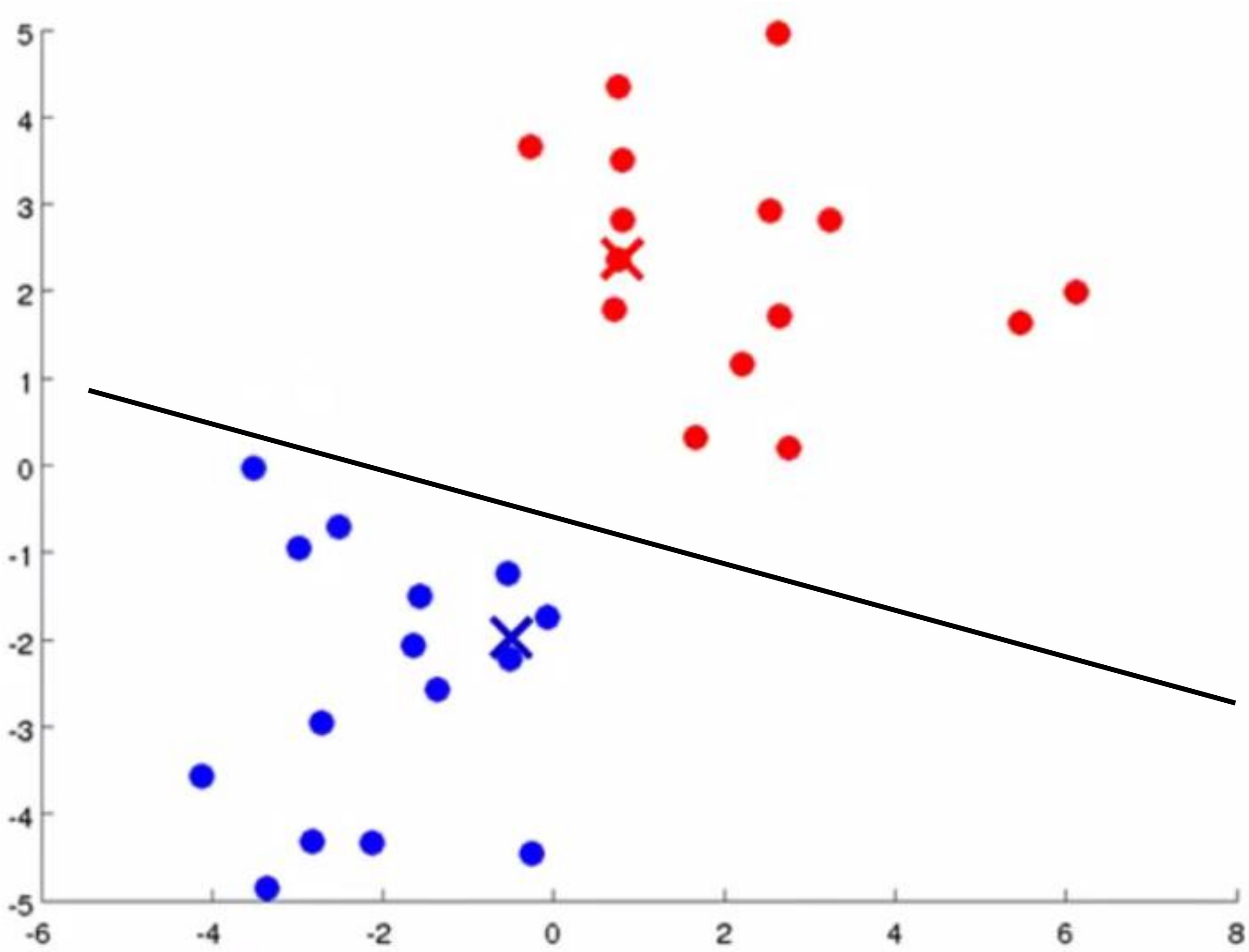
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

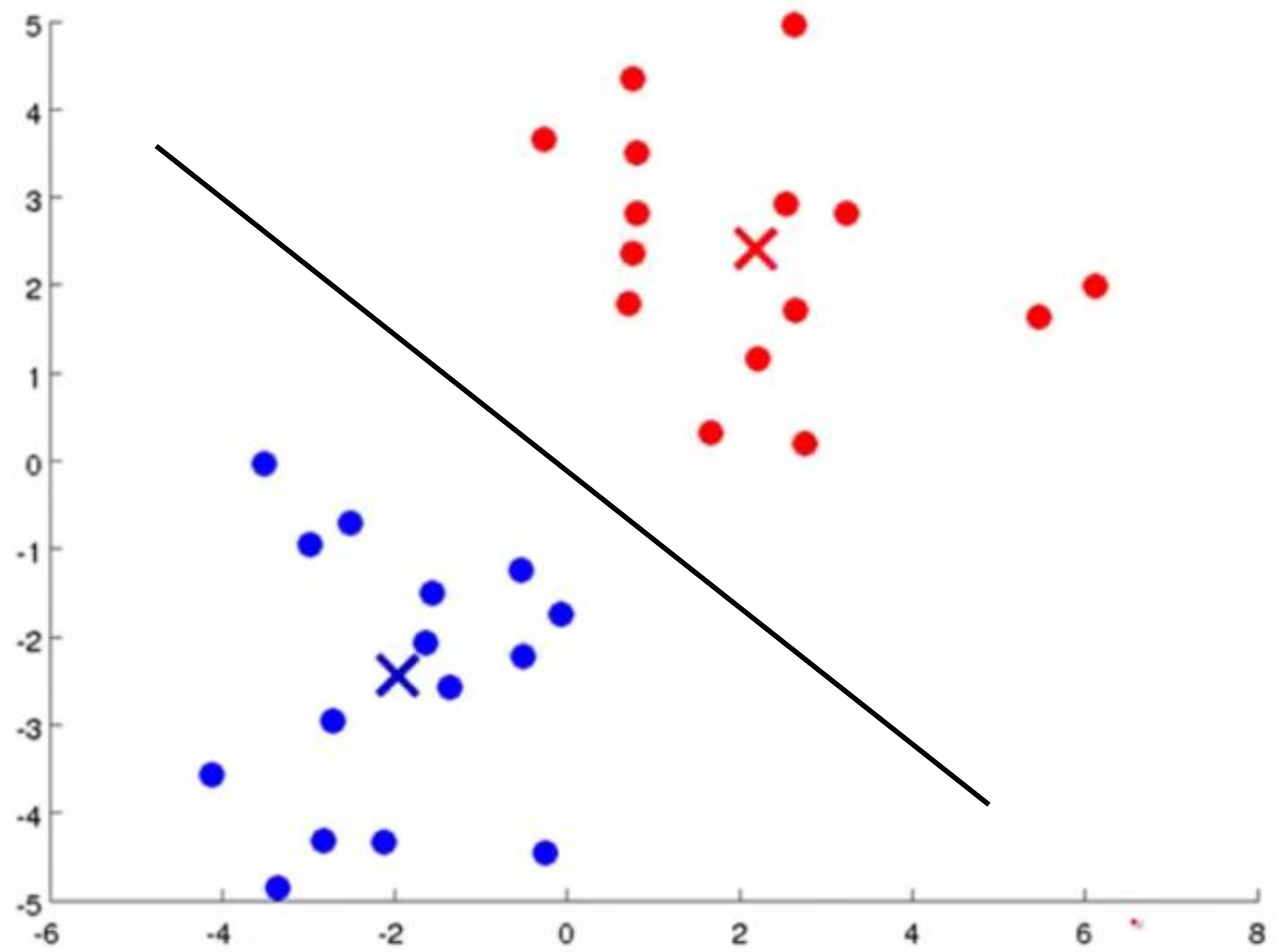

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8



5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8



5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

K-Means Algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1. Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships.
  4. If a convergence criterion is not met, or **if some clusters don't get any points**, go to 2.

Optimizing

$$\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}-\mu_{c^{(i)}}||^2$$

Stopping/Convergence Criterion

1. No (or minimum) re-assignments of data points to different clusters,
2. No (or minimum) change of centroids, or
3. Minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

– C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j

HOW DO WE EMPLOY DISTANCE IN A CLUSTER?

What do we mean by distance *between* clusters?

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: average distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- **Centroid: distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$**
- Medoid: distance between the medoids of two clusters, i.e., $dis(K_i, K_j) = dis(M_i, M_j)$
  - Medoid: one chosen, centrally located object in the cluster

Centroid, Radius, and Diameter of a Cluster (for Numerical Data Sets)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

ENGINEERING

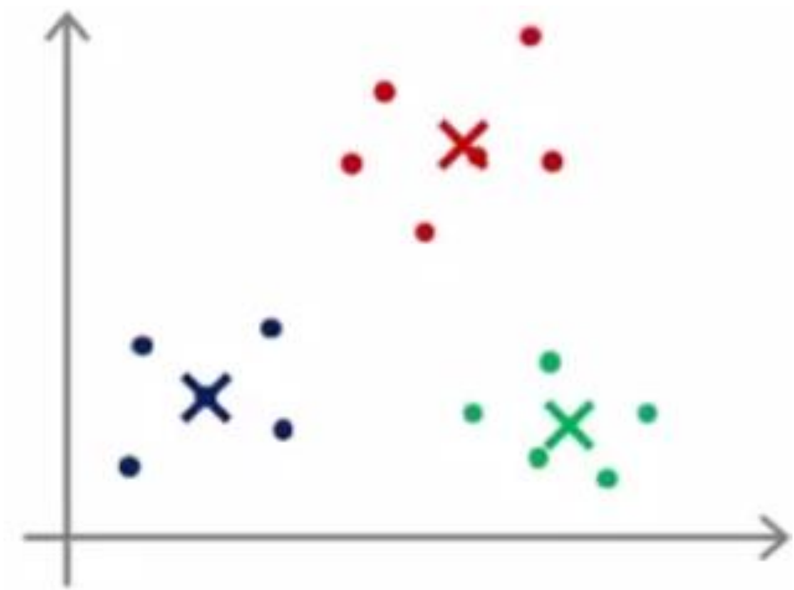
Stability Check of the Clusters

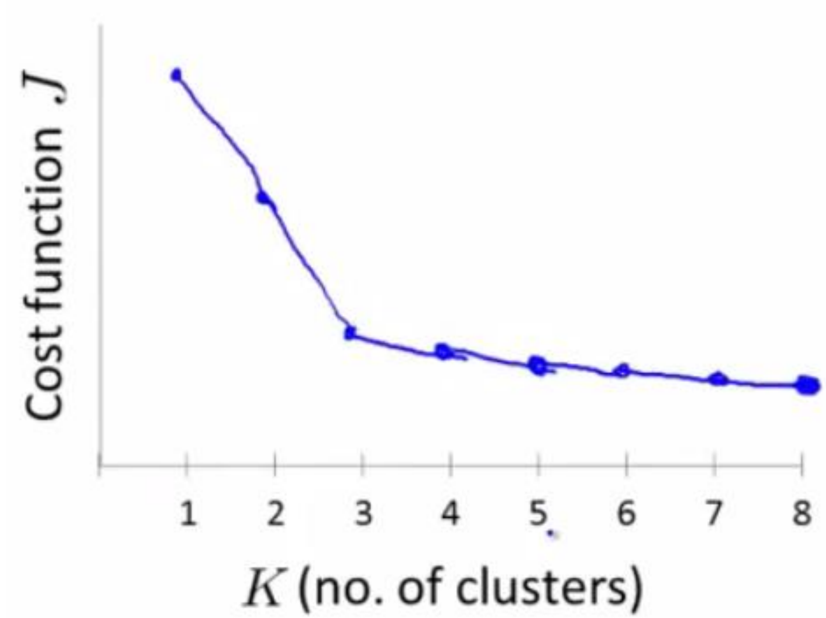
- To check the stability of the clusters take a random sample of 95% of records.

- Compute the clusters.

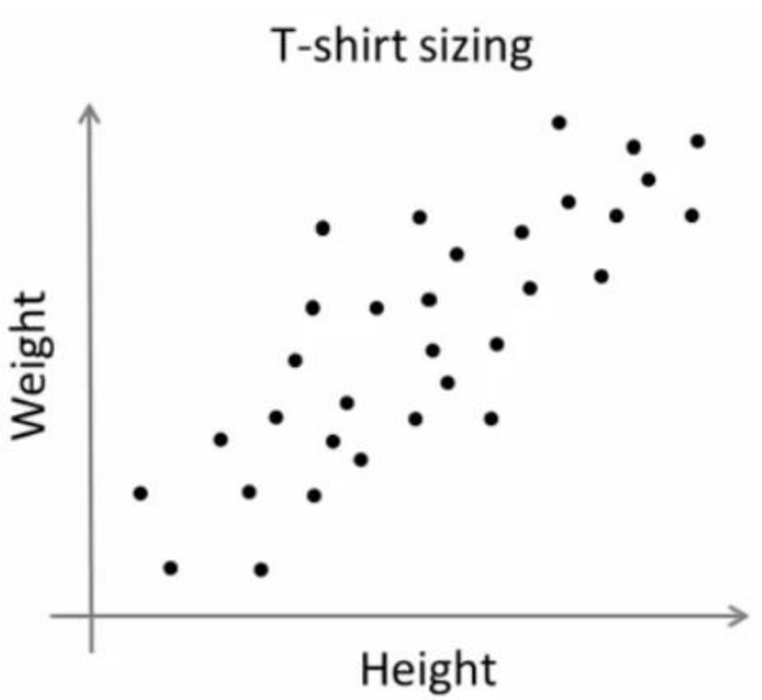
- If the clusters formed are very similar to the original, then the clusters are fine.

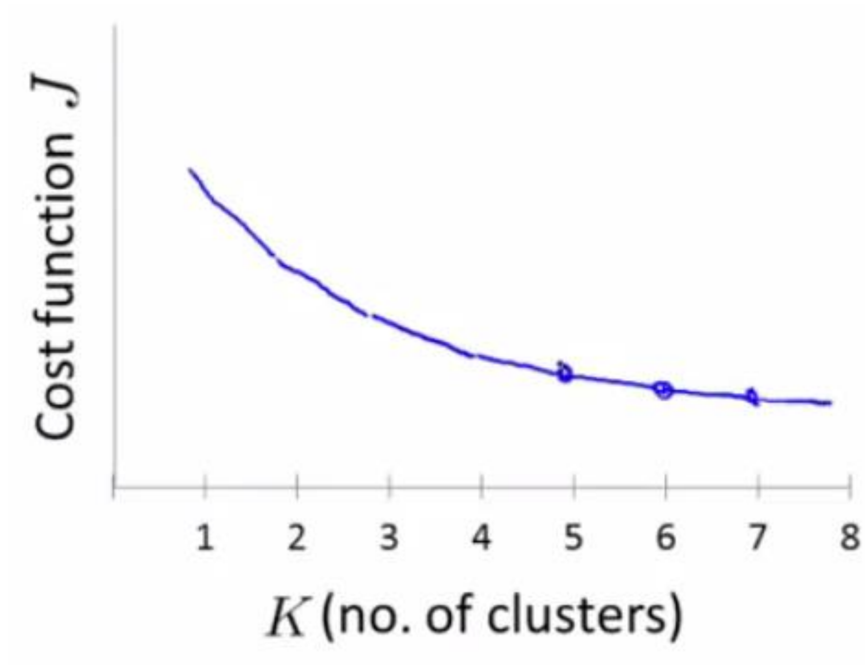
Linearly Clustered Data

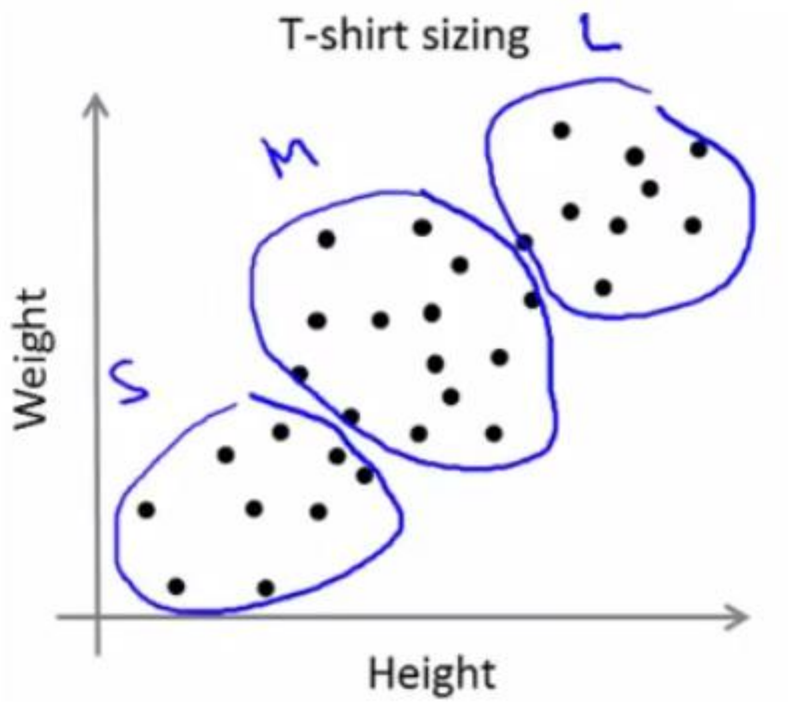


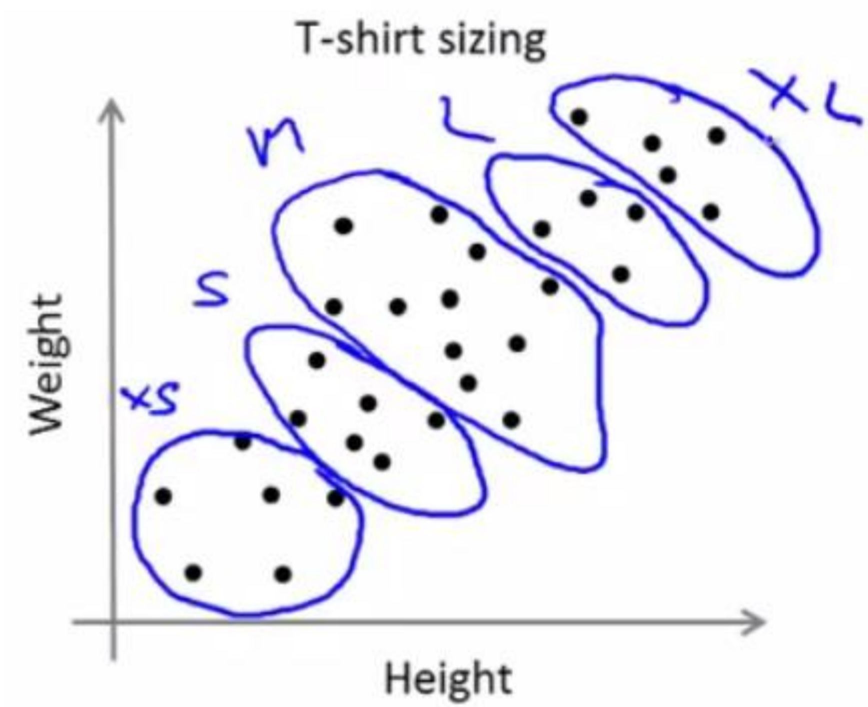


Linearly Separable but Merged





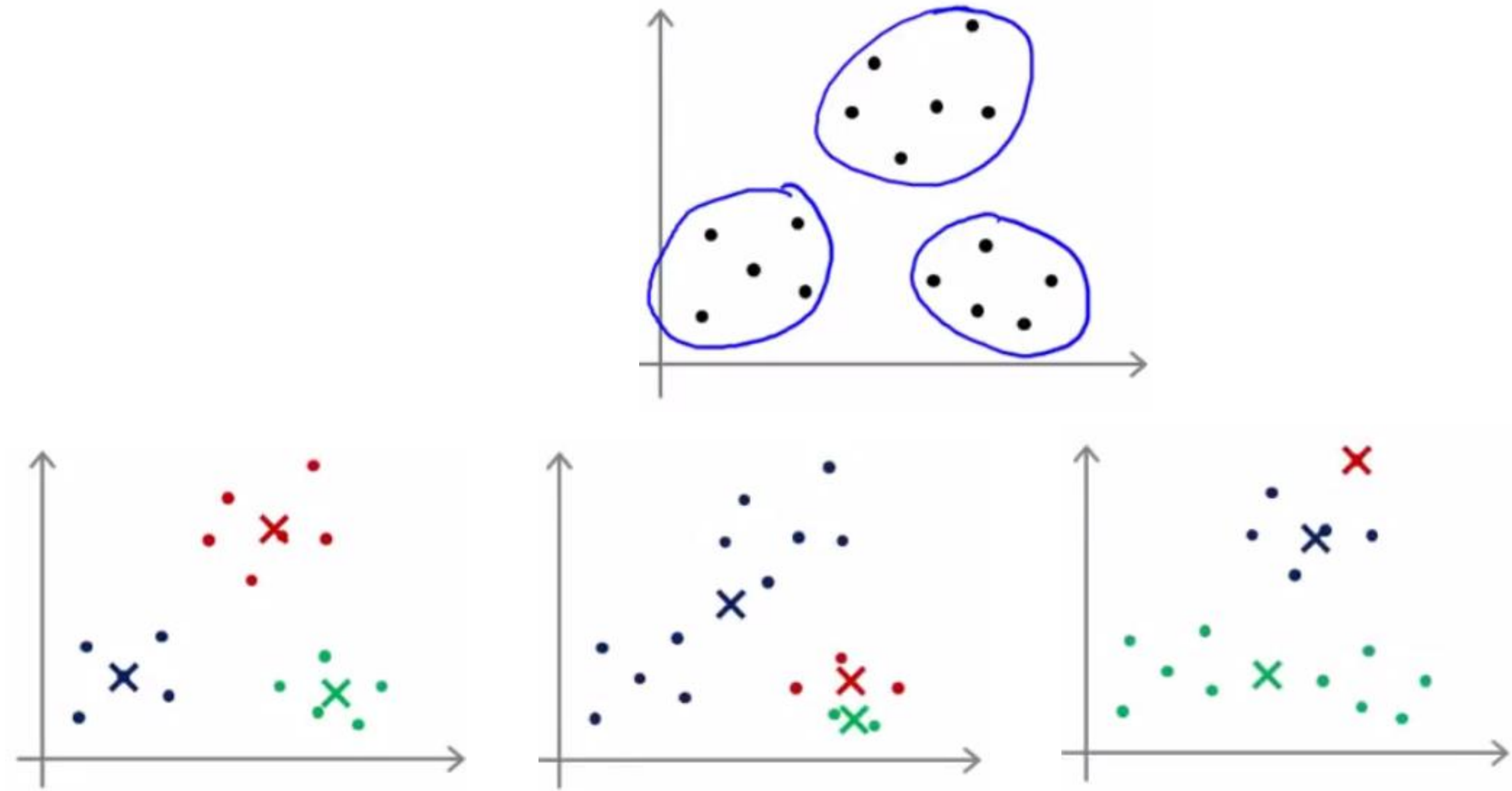

T-shirt sizing
L
M
S
Weight
Height


T-shirt sizing
XL
L
M
S
XS
Weight
Height

Linearly Separable

- Run 50-500 simulations for small k (2-10). For large k (100 or so), we can do 1-5 simulations
- Pick the one that gives the best S

Local Optima



What is the problem with K-Means?

- The k-means algorithm is sensitive to outliers!
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

What is the problem with Medoids?

- More robust than k-means, in the presence of noise and outliers, because a medoid is less influenced by outliers or other extreme values than a mean

- Works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

where n is # of data,k is # of clusters

K-Means vs. Hierarchical

- Flat clustering produces a single partitioning
- Flat clustering needs the number of clusters to be specified
- Flat clustering is usually more efficient run-time wise

- Hierarchical Clustering can give different partitionings depending on the level-of-resolution we are looking at
- Hierarchical clustering doesn’t need the number of clusters to be specified
- Hierarchical clustering can be slow (has to make several merge/split decisions)

- http://www.naftaliharris.com/blog/visualizing-dbscan-clustering/
- http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

MATRICES AND PCA

Matrix is very flexible representation

- Several different physical quantities can be represented as matrices
  - Transformations
  - Data
  - States
  - Relationships & graphs

MATRIX AS TRANSFORMATIONS

System of equations: Row view

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1\\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2\\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{aligned}
$$

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}
\overbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}^{\text{Vectors}}
$$

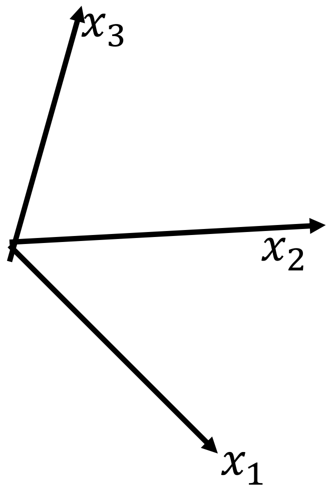
MATRIX AS COORDINATE AXES

System of equations: Column view

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Type equation here.

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} + x_3 \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$



An alternate view

- Row view gives equations

- Column view gives coordinate axes

Inner products

- W is a nonsingular nXn matrix.
- The inner product of 2 n dimensional column vectors x and y

$$< X, Y >_W = (WX).(\overline{WY})$$

When W= I

$$< X, Y > = X.\bar{Y}$$

Norms

$$The\ inner\ product\ generated\ norm\ \|X\|_W = \sqrt{< X, \bar{X} >_W}$$

$$Eucledean\ or\ l_2 norm\ \|X\|_2 = \sqrt{X.\bar{X}}$$

$$The\ l_1 norm\ \|X\|_1 = |x_1| + |x_2| + |x_3| + \cdots + |x_n|$$

$$The\ l_\infty norm\ \|X\|_\infty = \max(|x_1|, |x_2|, |x_3|, \ldots |x_n|)$$

$$The\ l_p norm\ \|X\|_p = (|x_1|^p + |x_2|^p + |x_3|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

Frobenius norm of a matrix is Eucledean version

$$\|A\|_F = \sqrt{\sum\sum |a|_{ij}{}^2}$$

Orthogonal vectors

- Inner product is zero (so, w.r.t one matrix they may be orthogonal, w.r.t other they may not be)
- Gram Schmidt orthogonalization
  - Every finite set of linearly independent vectors can be combined to create same number of orthogonal vectors

Orthogonal matrices

- Formed by orthonormal vectors (unit orthogonal)

$$Q^T Q = Q Q^T = I$$

Transpose is the inverse

Vector spaces

- Vectors that are linearly independent of each other. In n dimensions, if we have n linearly independent vectors, we can span the entire space
- These vectors are called bases

4 Sub spaces of a matrix

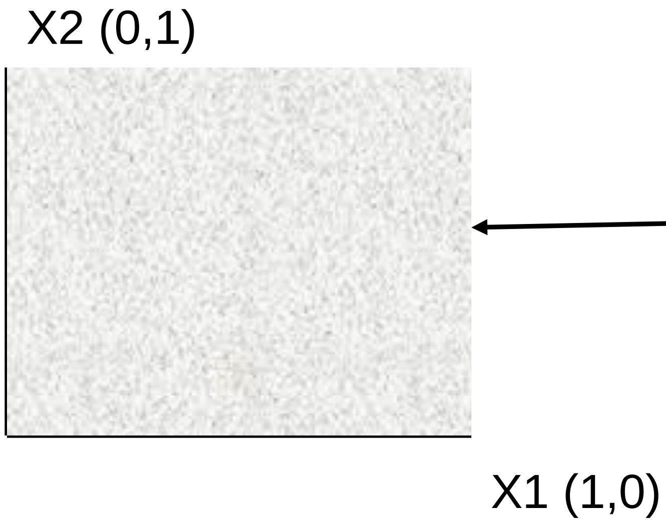
- For mXn matrix
  - Column space: $C(A)$ in R^n
  - Null Space: $N(A)$ in R^n (Solution of $Ax=0$)
  - Row Space: $C(A^T)$ in R^m
  - (left) Null Space of A^T: $N(A^T)$ in R^m

Rank of a matrix

- Linear independence: How many attributes are dependent on others (can be expressed as linear combinations)

- The number of linearly independent rows/columns is the rank

Determinant of a matrix: Cross product of column vectors



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

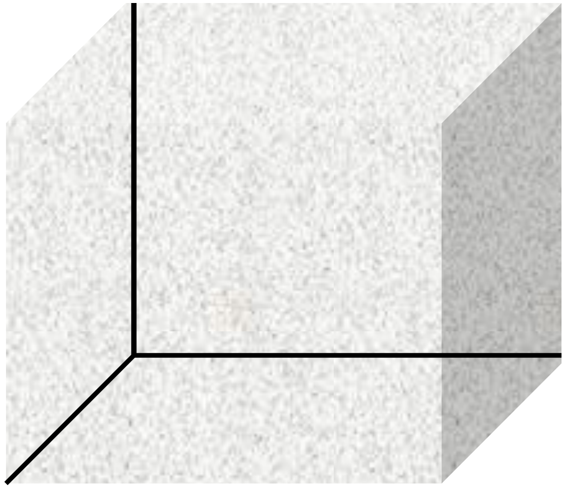
Determinant is a single number representation of the matrix.

Geometrically, it is area in 2 dimensions

It is signed. If we consider $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, the\ area\ is\ -1$

Determinant changes sign if rows or columns are interchanged

In higher dimensions

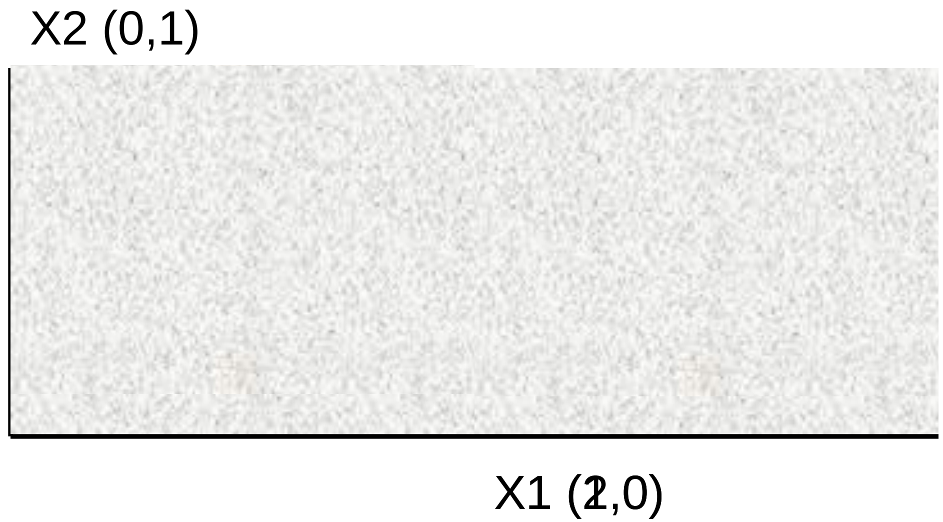


determinant is the
Signed volume of vectors comprised

Naturally

- If you multiply a column or row by a number

- determinant gets multiplied by the same

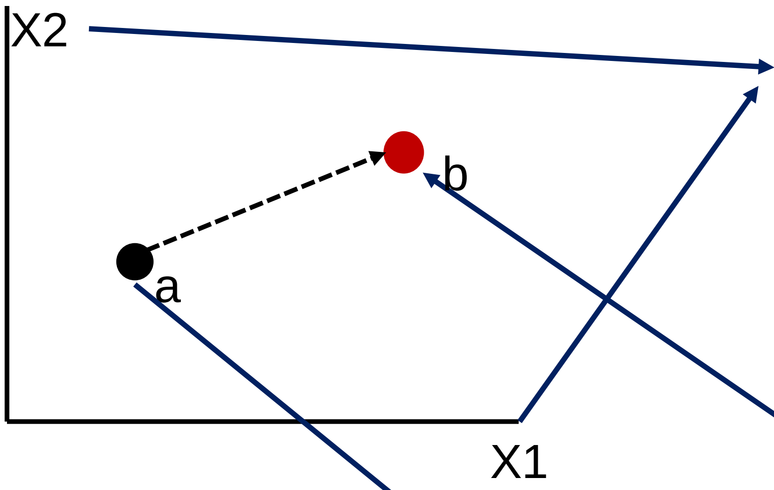


Naturally

- If both axes are linearly dependent (X2=k.X1), then the determinant is zero

MATRIX AS TRANSFORMATION ENGINE

Matrix as a transformation on a vector



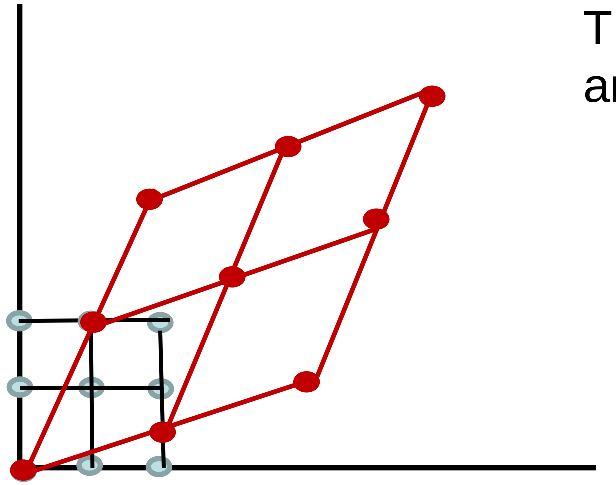
Coordinate system (matrix) = $[x_1 \quad x_2]$ *or* $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$\begin{bmatrix} x_{11} & x_{21} \\ x_{21} & x_{22} \end{bmatrix}_{[2X2]} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}_{[2X1]} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}_{[2X1]}$$

Transformation matrix

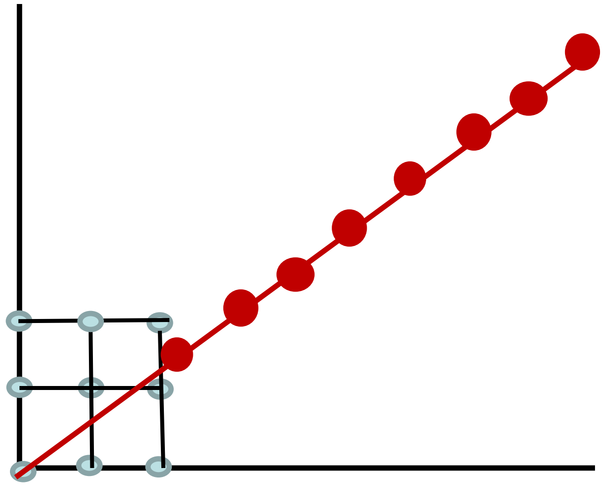
A MATRIX OPERATES ON A VECTOR AND TRANSFORMS IT TO ANOTHER VECTOR

Matrix transformation on spaces



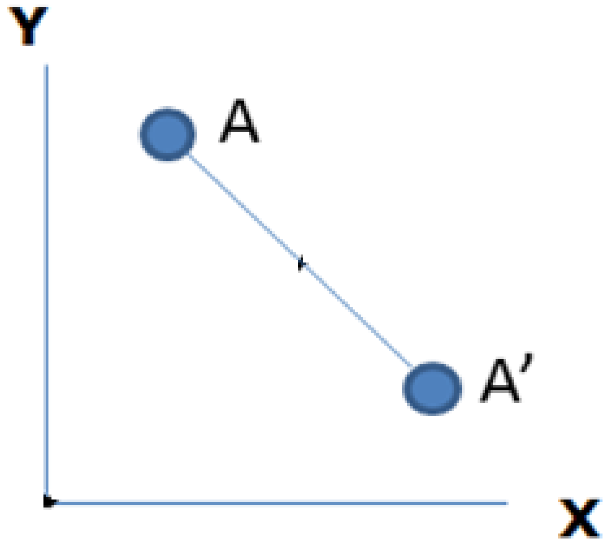
This matrix is stretching, rotating and skewing the space

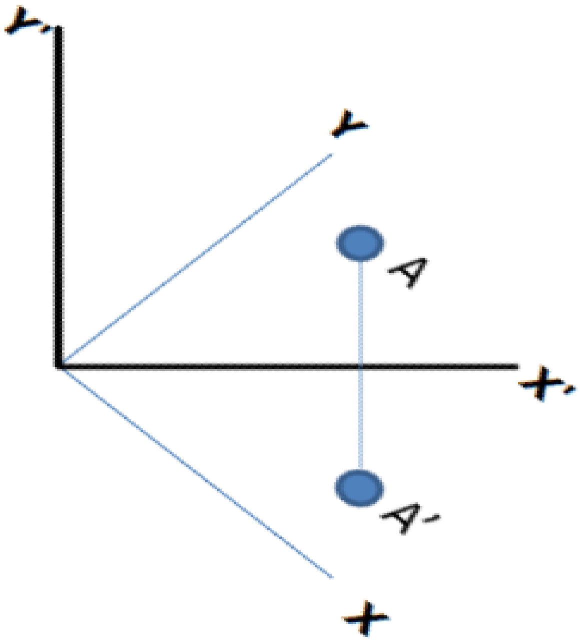
If determinant is zero for a transformation matrix, (they are singular)



Similarly, a low ranked transformation matrix takes the grid to a low dimensional space

Transformation operation is dependent on the basis!!!



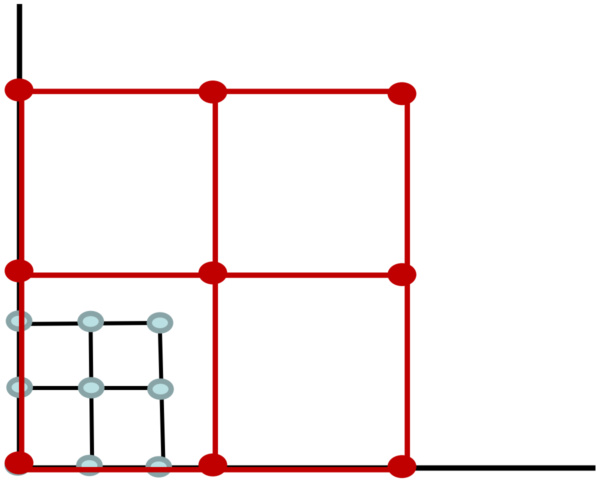


$$A = \frac{1}{\|\vec{l}\|^2}\begin{bmatrix} l_x^2 - l_y^2 & 2l_x l_y \\ 2l_x l_y & l_y^2 - l_x^2 \end{bmatrix}$$

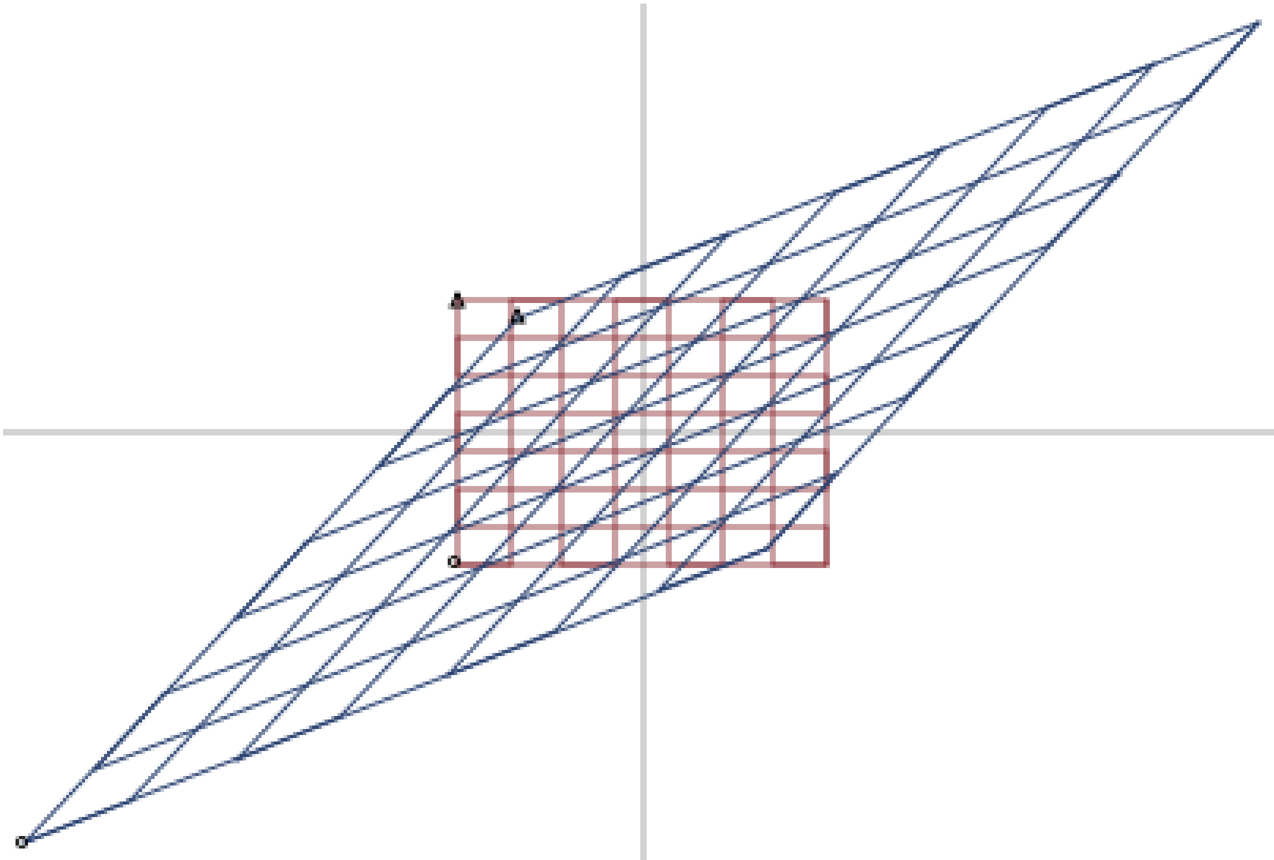
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

AHA! BY CHANGING THE AXES, WE CAN MAKE A COMPLEX TRANSFORMATION SIMPLE

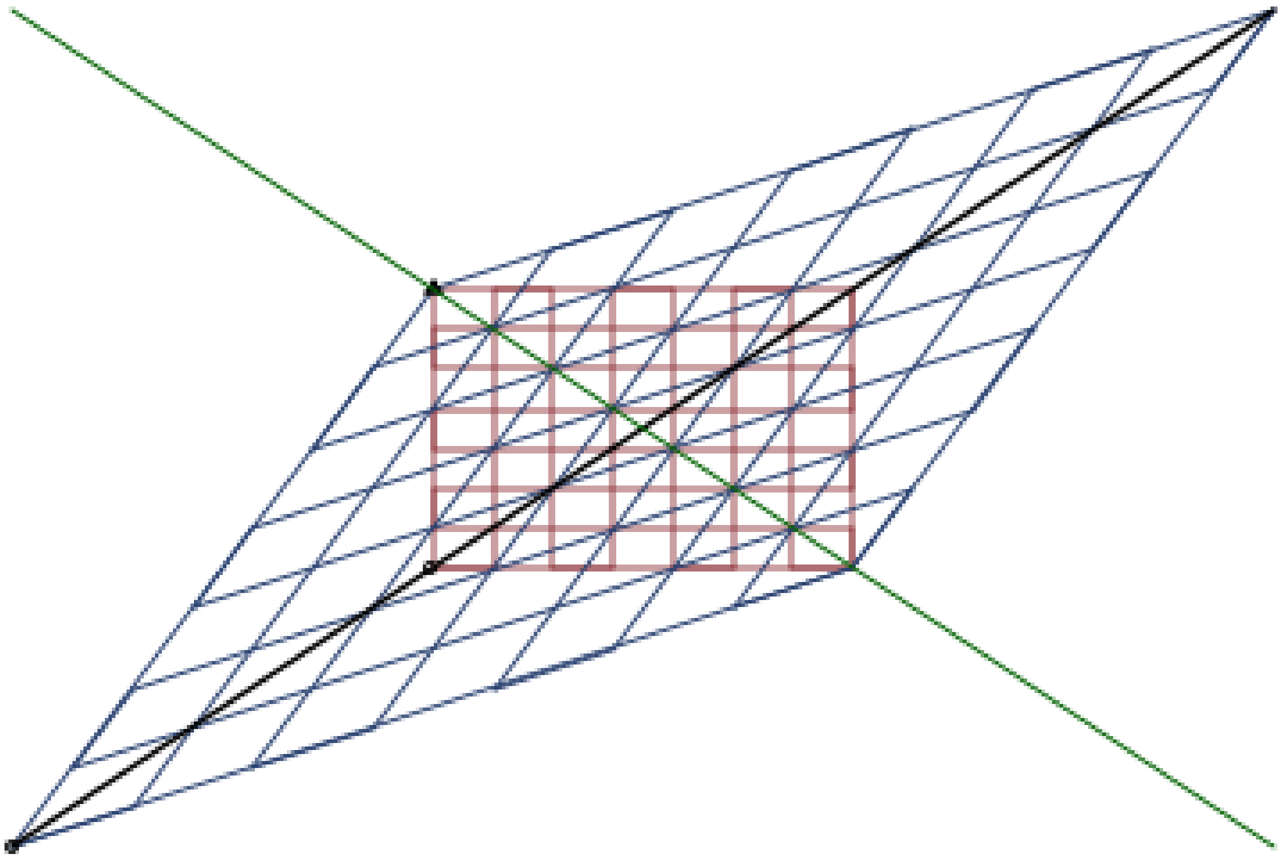
Can we find a basis where a transformation is changed to a purely stretch



A transformation matrix that stretches, rotates and skews



Eigen vectors



Eigen vectors mathematics

- The eigenvectors and eigenvalues of matrix **A** are defined to be the nonzero **x** and λ values that solve

- $\mathbf{Ax} = \lambda\mathbf{x}$ **(A is just stretching)**

Characteristic equations

The characteristic equation of a nXn matrix A is
The nth degree polynomial equation

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

So

- Eigen vectors of a transformation is that basis where transformation is only a stretch

- Eigen values are the magnitude of stretch

Factorization

- Not much can be said about 1728

- But, a lot can be said about

$$2^6 X 3^3$$

A matrix can be factorized too

- The idea of factorization is to split a non-special matrices into special constituents
- A = LU (decomposition into lower and upper triangular matrices)
- Solving equations becomes easy (forward, backward substitutions)

Eigen decomposition (A factorization)

$$AQ = A[x_1\ x_2\ \ldots x_n] = [\Lambda_1 x_1\ \ \Lambda_2 x_2 \ldots \Lambda_n x_n]$$

$$= [x_1\ \ x_2 \ldots x_n]\begin{bmatrix} \Lambda_1 & 0 & 0 \\ 0 & \Lambda_2 & 0 \\ 0 & 0 & \Lambda_n \end{bmatrix} = Q\Lambda$$

$$A = Q\Lambda Q^{-1}$$

A is an nXn square matrix with linearly independent eigen vectors
Q is the eigen vector matrix where each vector corresponds to one eigen value
Λ is a diagonal matrix formed by eigen values

An $n \times n$ matrix A is diagonalizable over the field F if it has n distinct eigenvalues in F, i.e. if its characteristic polynomial has n distinct roots in F; however, the converse may be false. (unit matrix)

Powers

$$\mathrm{A} = \mathrm{Q}\Lambda Q^{-1}$$

$$A^2 = \mathrm{Q}\Lambda Q^{-1}\mathrm{Q}\Lambda Q^{-1} = \mathrm{Q}\Lambda Q^{-1}$$

or

$$Ax = \Lambda x; A^2 x = A\Lambda x = \Lambda A x = \Lambda^2 x$$

$$A^k = \mathrm{Q}\Lambda^k Q^{-1}$$

- Any two matrices connected with the above relation are called similar matrices.
- So, diagonal matrix is a similar matrix in diagonal form.
- Similar matrices have same eigen values

Properties of Eigen values

- The sum of eigen values is equal to trace (sum of the main diagonal elements)
- A matrix and its transpose have same eigen values
- Eigen values of L and U are elelments of its main daigonals

Hermitian matrices

- Hermitian transpose is a complex conjugate of a matrix.
- A normal matrix is where

$$AA^H = A^H A$$

A normal matrix has orthonormal eigen vectors and can be diagonolized

Hermitian matrices

- A matrix is Hermitian if it equals its Hermitian transpose

$$A = A^H$$

All real symmetric matrices are Hermitian and hence are normal

Variance-covariance matrix

$$\begin{bmatrix} {\sigma_a}^2 & \rho_{ab} & \rho_{ac} \\ \rho_{ba} & {\sigma_b}^2 & \rho_{bc} \\ \rho_{ca} & \rho_{cb} & {\sigma_c}^2 \end{bmatrix}$$

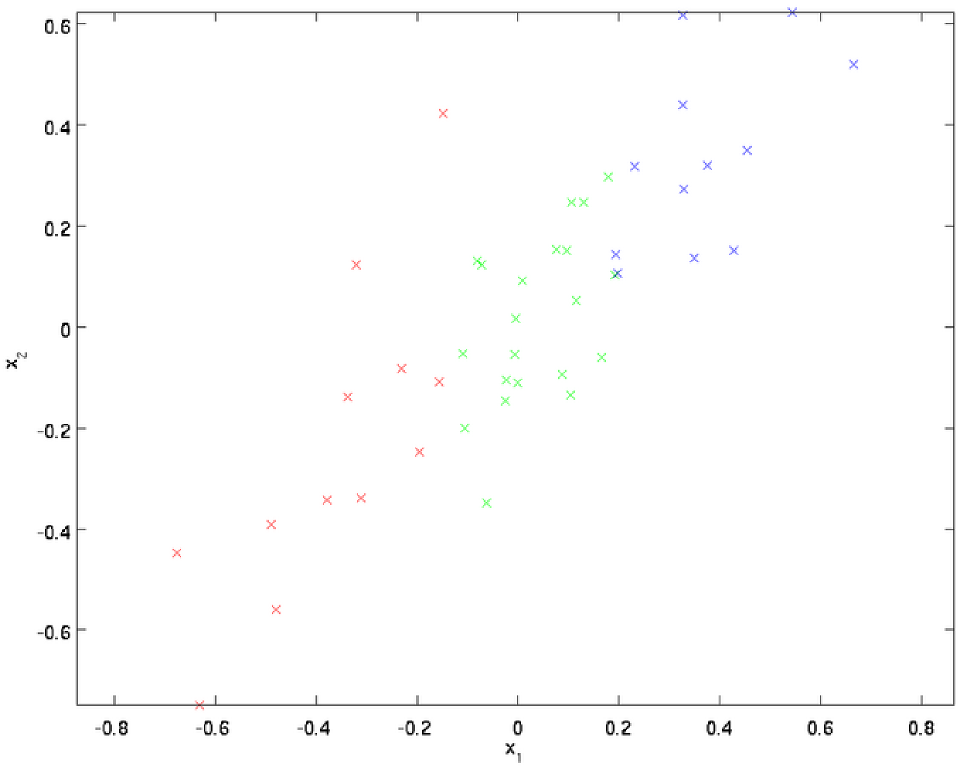
This is how data is spread in each axis

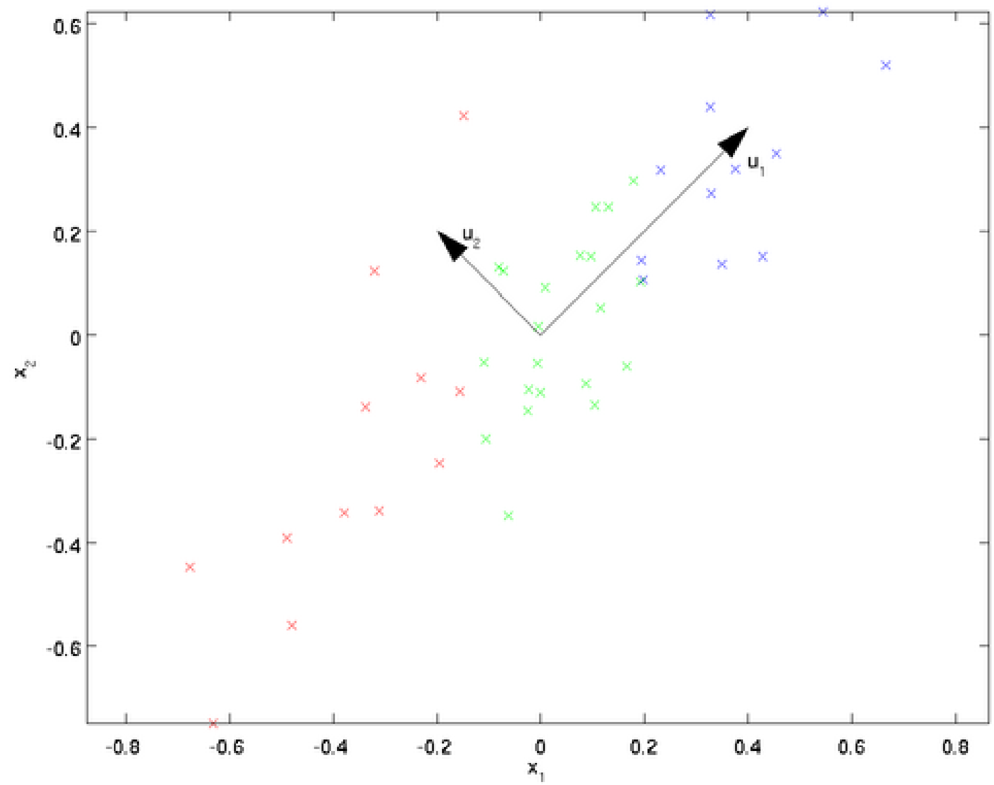
It is Hermitian and Normal and hence Diagonalizable with Orthonormal eigen vectors

Now find the basis where it is just a stretch

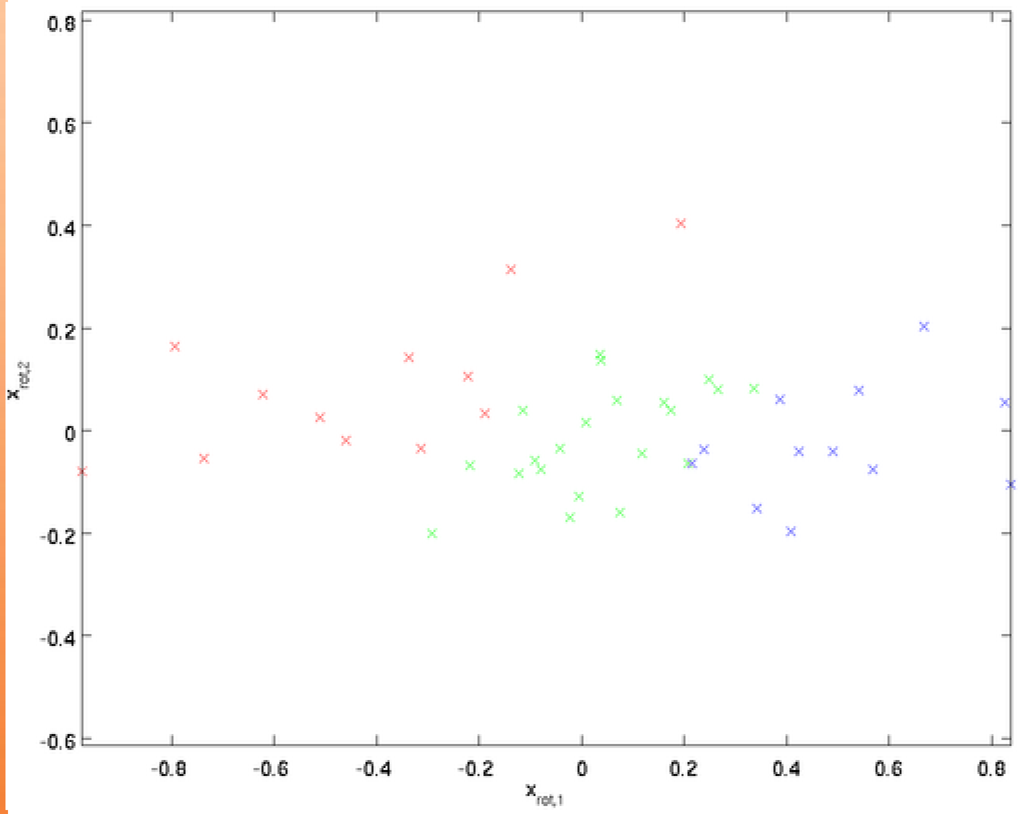
$$\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

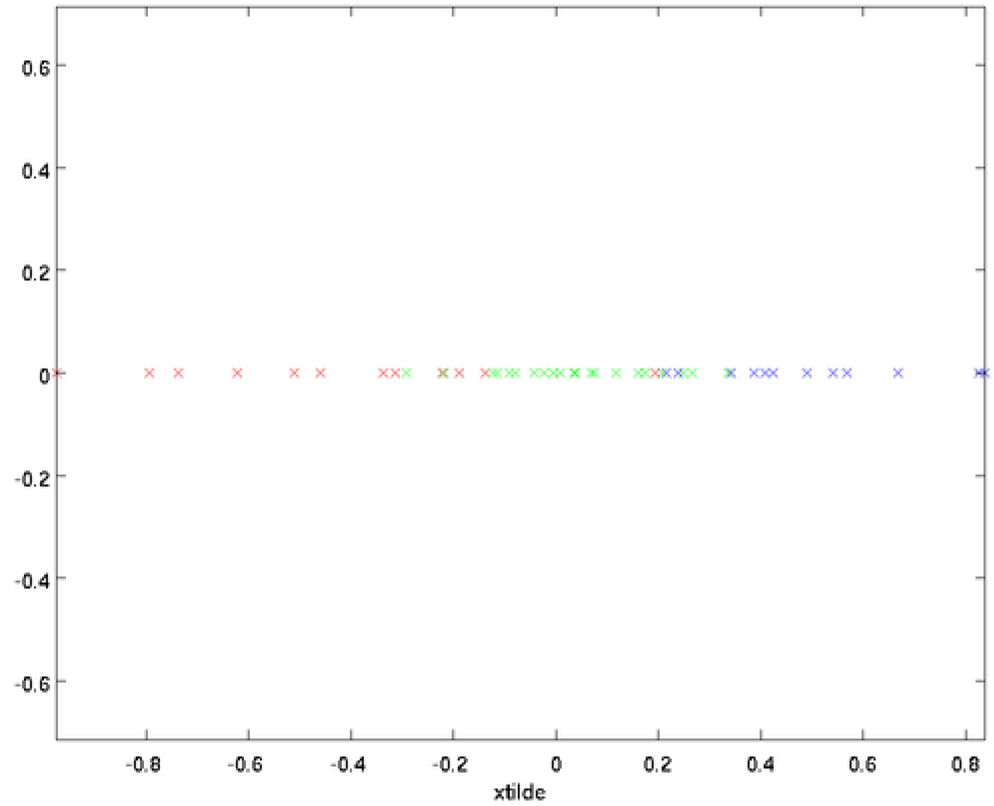
Eigen vectors are the basis





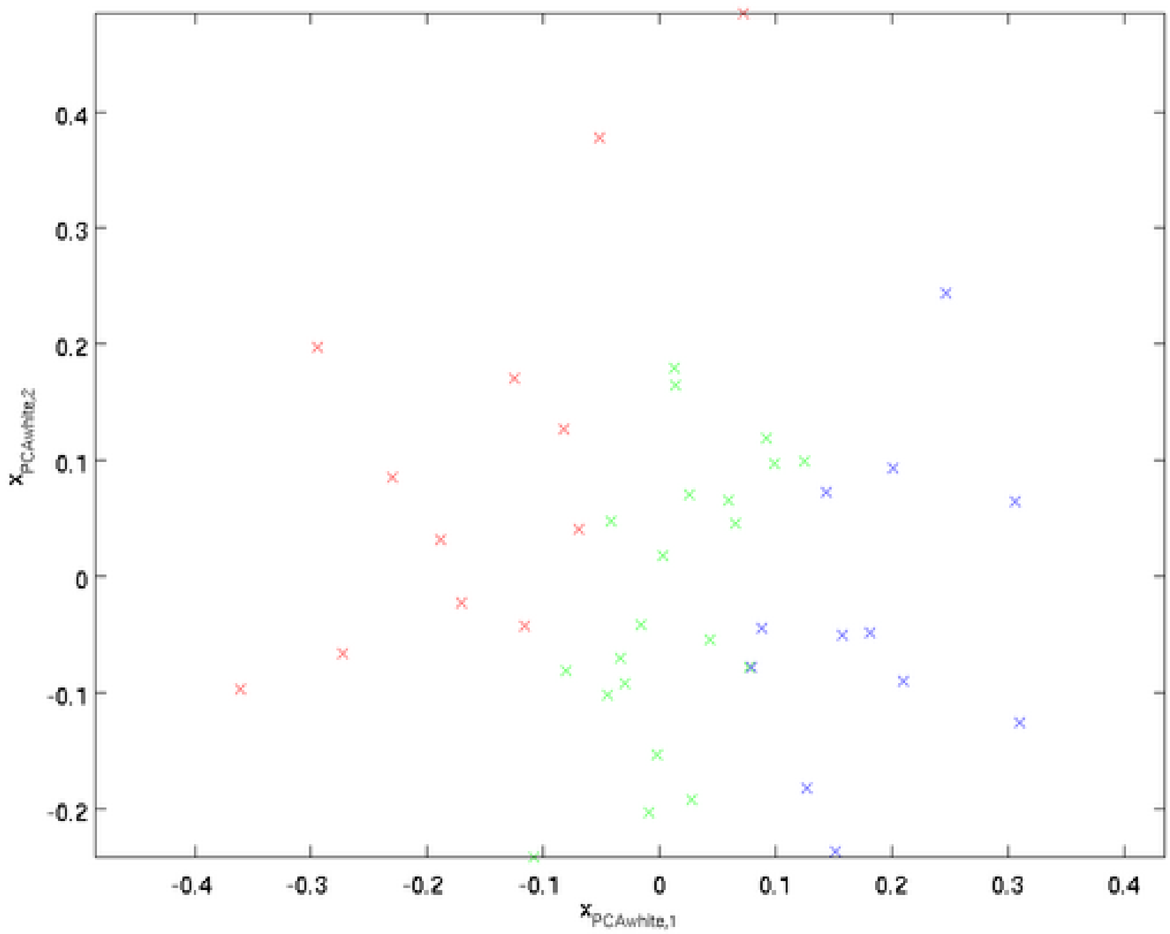
In the new space (and reduced dimensionality)





Whitening

$$x_{\mathrm{PCAwhite},i} = \frac{x_{\mathrm{rot},i}}{\sqrt{\lambda_i}}$$



Dimensionality reduction

- Drop dimensions where eigen values are small because the variance (or stretch) in those axes is small

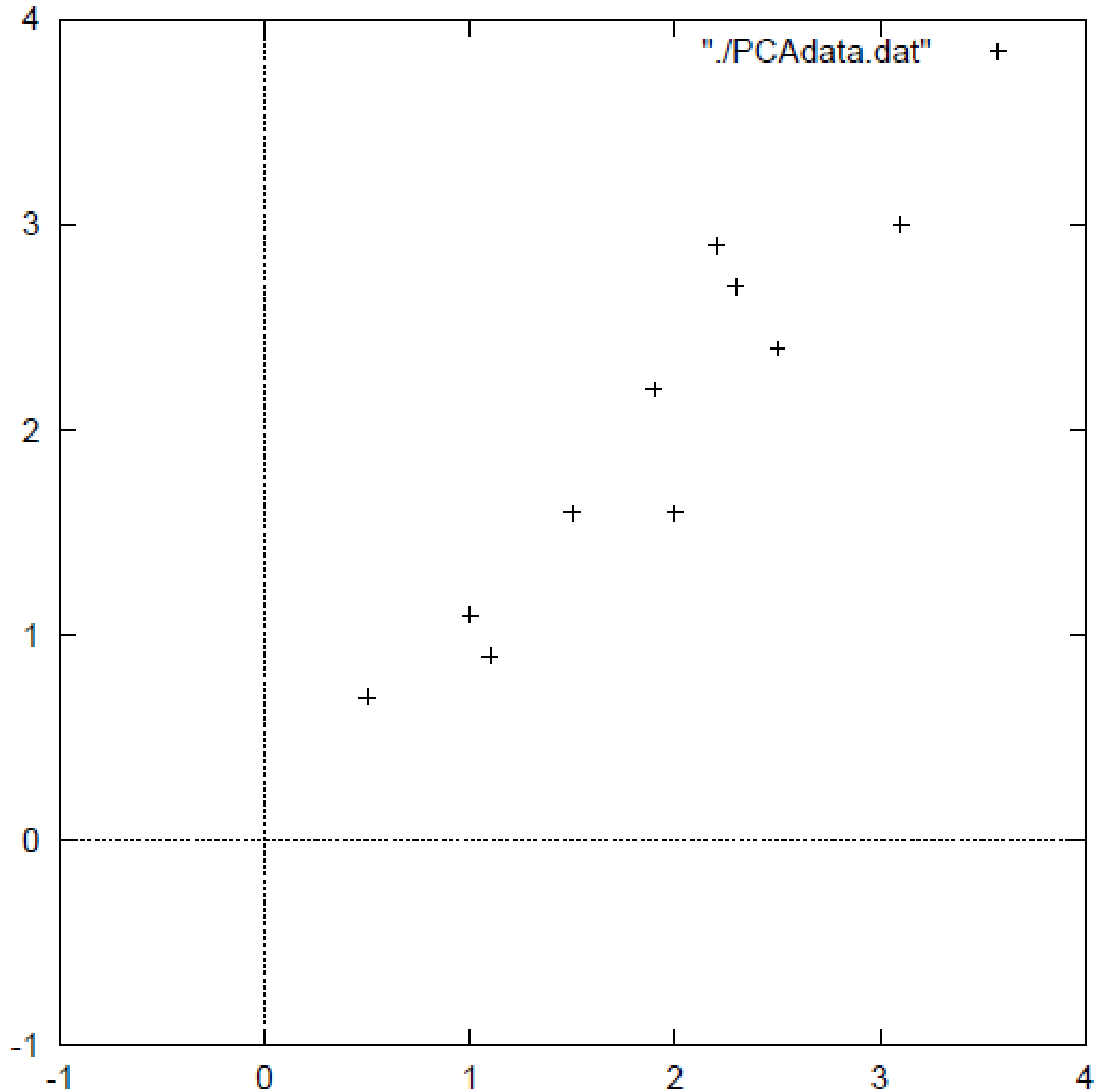
Process

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 =

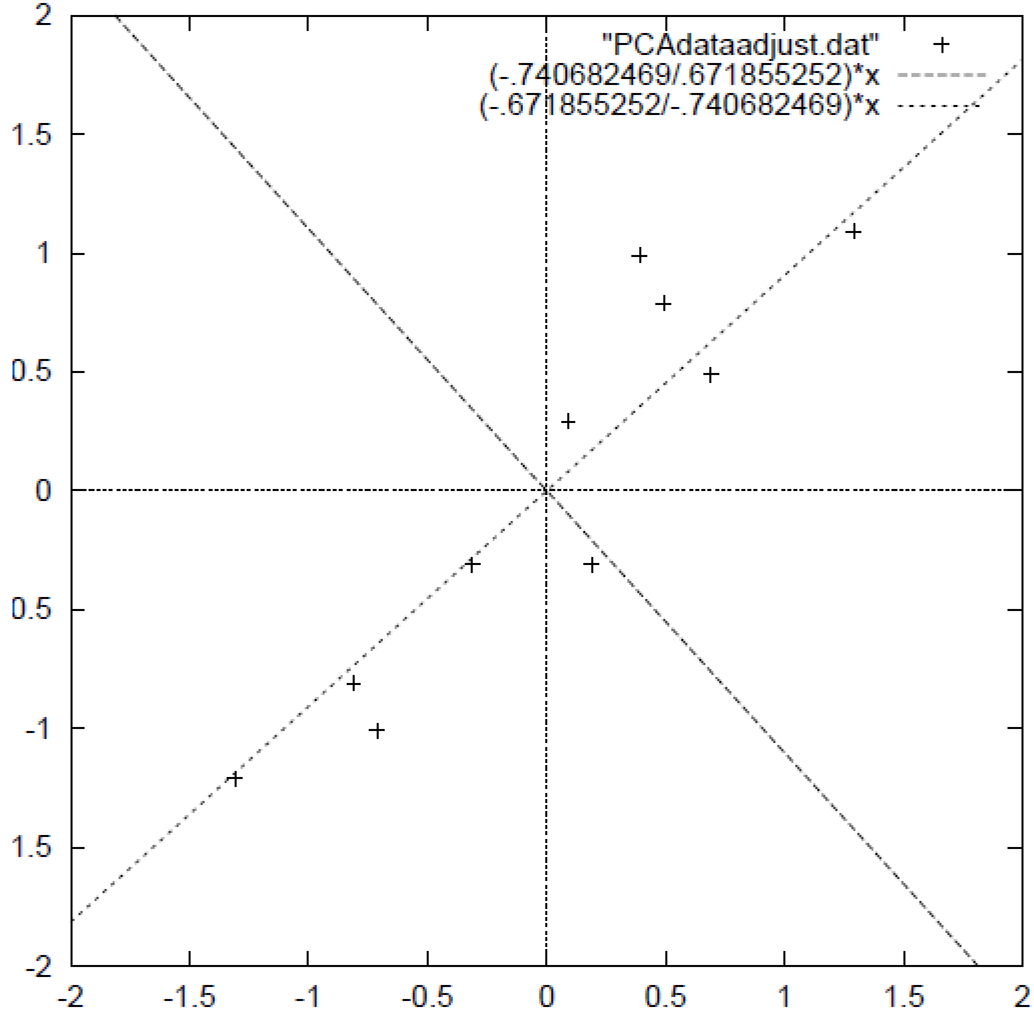
x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01


"./PCAdata.dat"
4
3
2
1
0
-1
-1
0
1
2
3
4

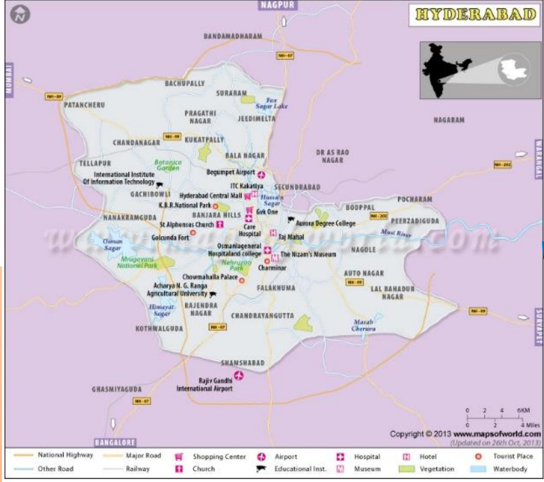
$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

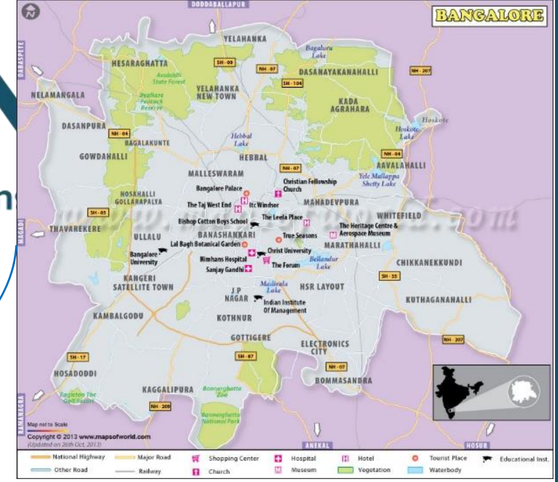
$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$


"PCAdataadjust.dat"
(-.740682469/.671855252)*x
(-.671855252/-.740682469)*x
2
1.5
1
0.5
0
0.5
-1
1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

R implementation of PCA





HYDERABAD

Office and Classrooms
Plot 63/A, Floors 1&2, Road # 13, Film Nagar,
Jubilee Hills, Hyderabad - 500 033
+91-9701685511 (Individuals)
+91-9618483483 (Corporates)

BENGALURU

Office
Incubex, #728, Grace Platina, 4th Floor, CMH Road,
Indira Nagar, 1st Stage, Bengaluru – 560038
+91-9502334561 (Individuals)
+91-9502799088 (Corporates)

Classroom
KnowledgeHut Solutions Pvt. Ltd., Reliable Plaza,
Jakkasandra Main Road, Teacher's Colony, 14th Main
Road, Sector – 5, HSR Layout, Bengaluru - 560102

Social Media

Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	http://www.youtube.com/InsofeVideos
SlideShare:	http://www.slideshare.net/INSOFE
LinkedIn:	http://www.linkedin.com/company/international-school-of-engineering