Unsupervised Learning

Kai Chen, CSE, HKUST

MSBD 5001 Lecture 4

Previous Topics: Supervised Learning

- Last two weeks: supervised learning
- Observe a set of data with features X, and the output Y for each data point.
- Goal: predict the outcome variable Y with given X
  - Classification: Predict a categorical result.
  - Regression: Predict a numerical value.

Unsupervised Learning

- Another important class of machine learning methods: Analyze the structure of the data using feature X.
- Supervised: Use features X to predict labels Y
- Unsupervised: Only requires features, don't deal with labels.
- Examples:
  - 1. Clustering: Given a dataset divide it into meaningful groups. A data point is more similar with points in the same group, compared to those in different groups.

Unsupervised Learning Cont'd

- Examples:
  - 2. Dimensionality Reduction: I have a dataset of extremely high dimension of features (e.g., images), can I represent them with a lower dimension?
  - 3. Ranking: I have a dataset represented as a graph, each data point is a node, and their relationship are edges, e.g., the World Wide Web, can I rank the importance of the data points?
- Today we introduce some unsupervised learning algorithms for these problems.

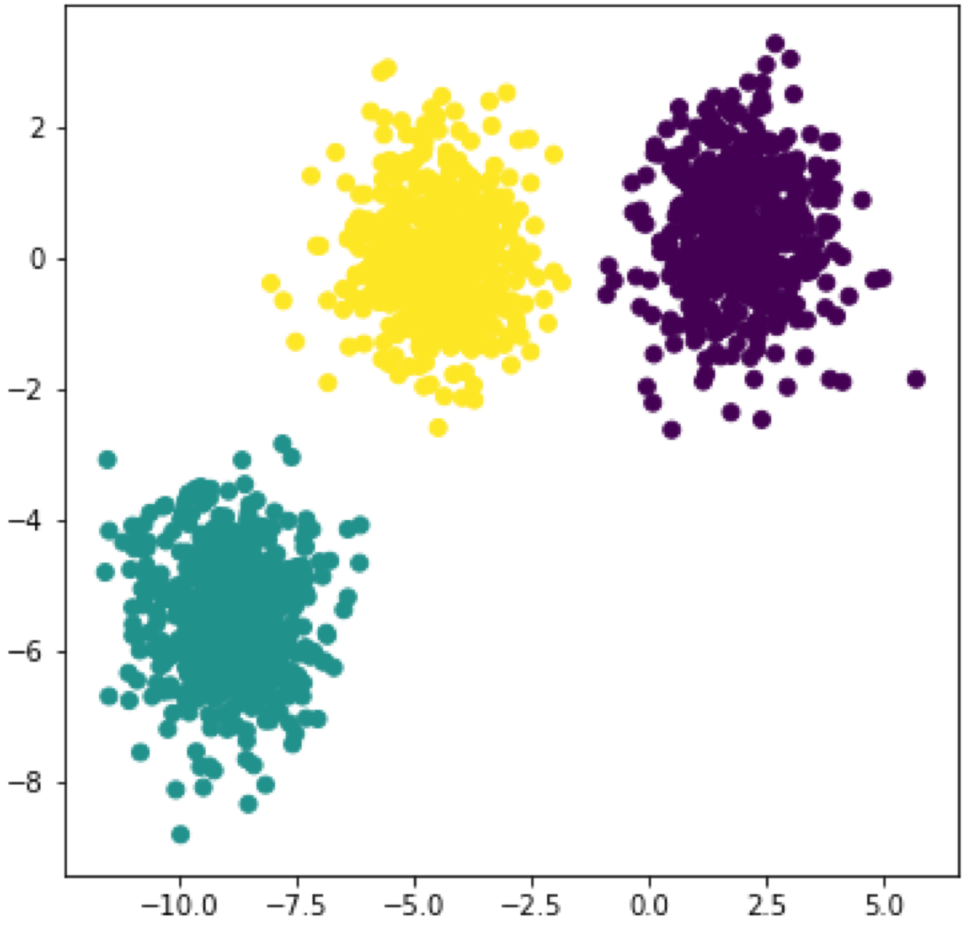
Outline

- Clustering:
  - K-means clustering
  - Hierarchical clustering
- Dimensionality Reduction
  - Principle Component Analysis
- Ranking
  - Google's PageRank

What is Clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - Objects in the cluster should be more similar.
  - Objects in different clusters should be less similar.
- An important form of unsupervised learning

A Dataset with Clear Cluster Structure



- How would you design an algorithm to find the three clusters in this case?

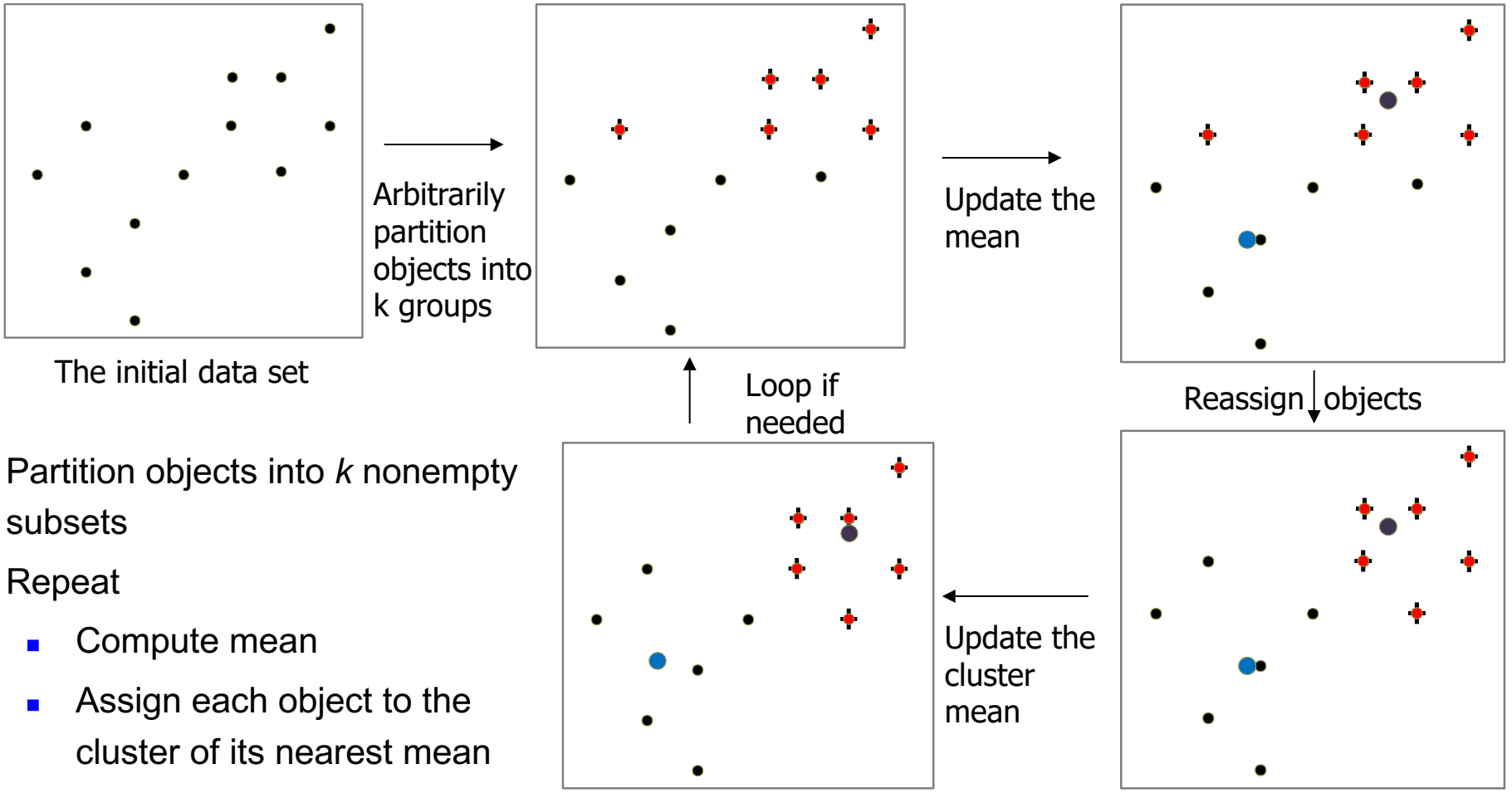
Intuitive Idea of Clustering

- Find a good similarity measure of points, assign similar points to the same cluster.
- Simpler version: Suppose clusters already exist, given a new data point, which cluster does it belong to?
- One possible answer: Find mean of the points in each cluster, the new point belong to the cluster whose mean is the closest to the point
- This is the principle idea of k-means clustering.

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - 1. Partition objects into *k* nonempty subsets
  - 2. Compute the mean point for every cluster at current partitioning
  - 3. Assign each object to the cluster with the nearest mean point
  - 4. Go back to Step 2, stop when the assignment does not change

An Example of *K-Means* Clustering (K=2)



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute mean
  - Assign each object to the cluster of its nearest mean
- Until no change

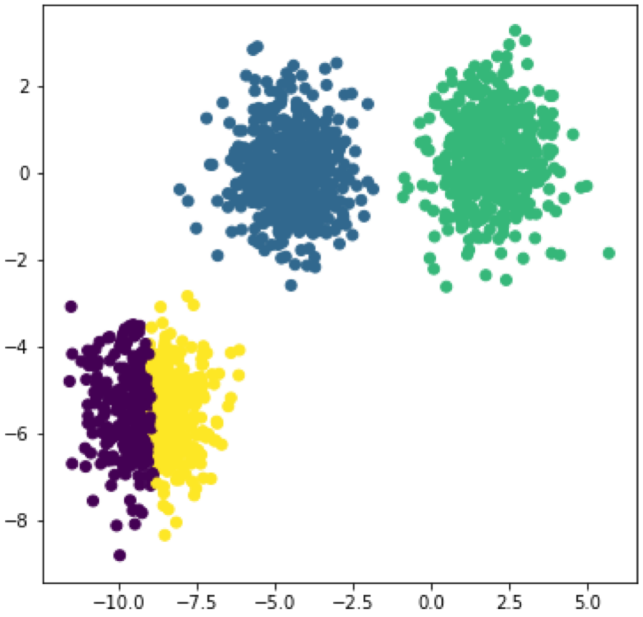
Practice Question

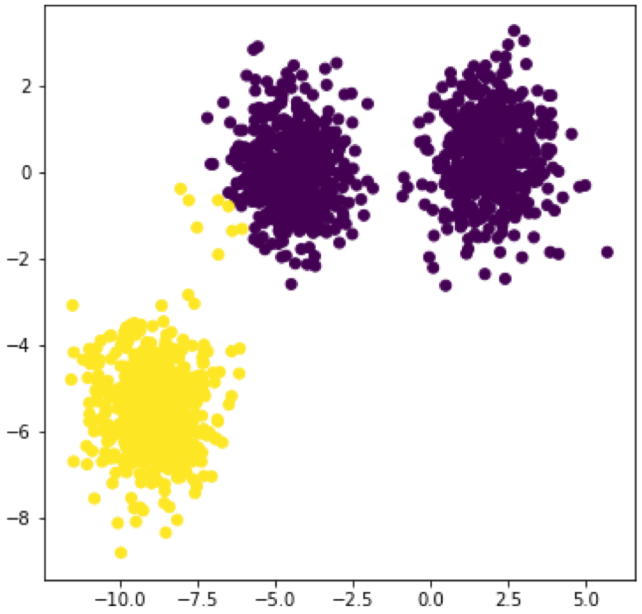
- Suppose you have 5 points in 1-D: {1,2,4,7,10}. Use k-means to cluster these points with k=2. Start with initial partition {1} and {2,4,7,10}. The distance is difference of coordinate on the axis.

No.	Cluster1	Cluster2	Mean1	Mean2
0	1	2,4,7,10	1	5.75
1	1,2	4,7,10	1.5	7
2	1,2,4	7,10	7/3	8.5
3	1,2,4	7,10	Terminates	

How to Select Value K?

- Both large K and small K can lead to bad results. Left K=4, right K=2.
- Didn't describe data well





Hierarchical Clustering

- A method of cluster analysis which seeks to build a hierarchy of clusters
- No need to decide number of clusters beforehand, but needs a termination condition

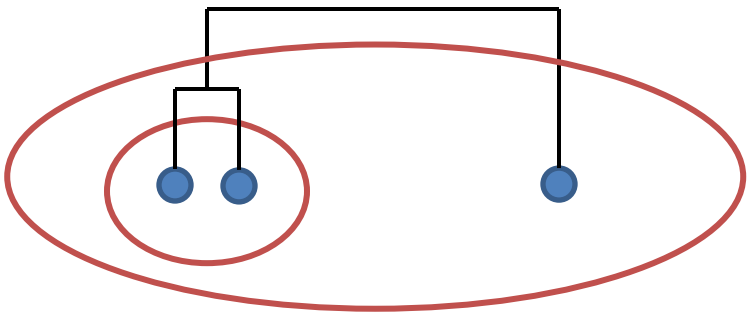
Hierarchical Clustering

- Two types: (1) agglomerative (bottom up), (2) divisive (top down)
- Agglomerative: two groups are merged if distance between them is less than a threshold
- Divisive: one group is split into two if intergroup distance is more than a threshold
  - Not discussed today
- Skip details of defining distance

Dendrograms

- Decompose data objects into a several levels of nested partitioning called a dendrogram
- Form a dendrogram:
  - Begin with every point “isolated”, i.e. in different clusters
  - Repeat: Among all possible merges, select the one of smallest distance
  - Until every point in the same cluster
- Cut the dendrogram: Clustering is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

Forming a dendrogram



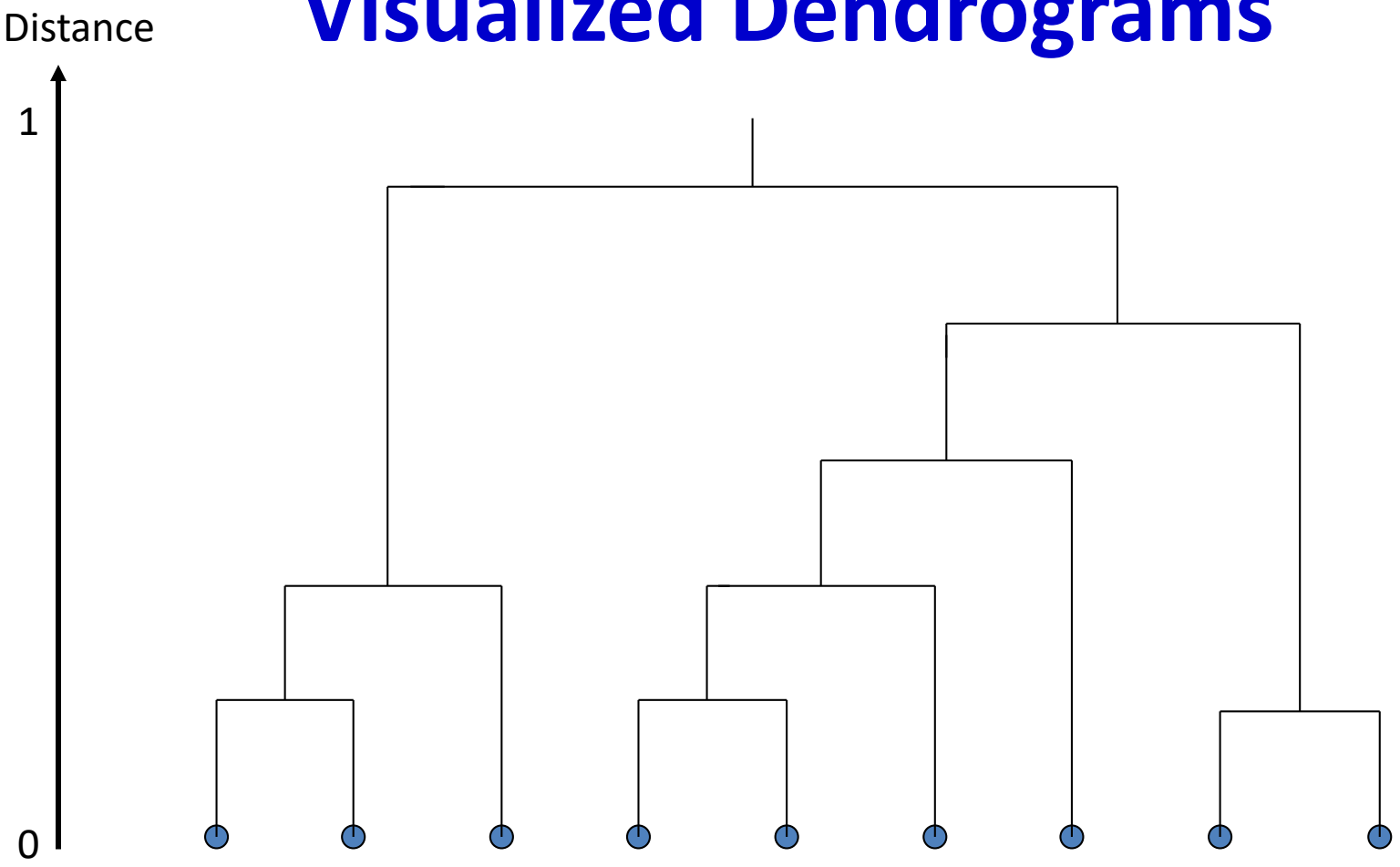
Begin with every point “isolated”, i.e. in different clusters

Repeat: Among all possible merges, select the one of smallest distance

Until every point in the same cluster

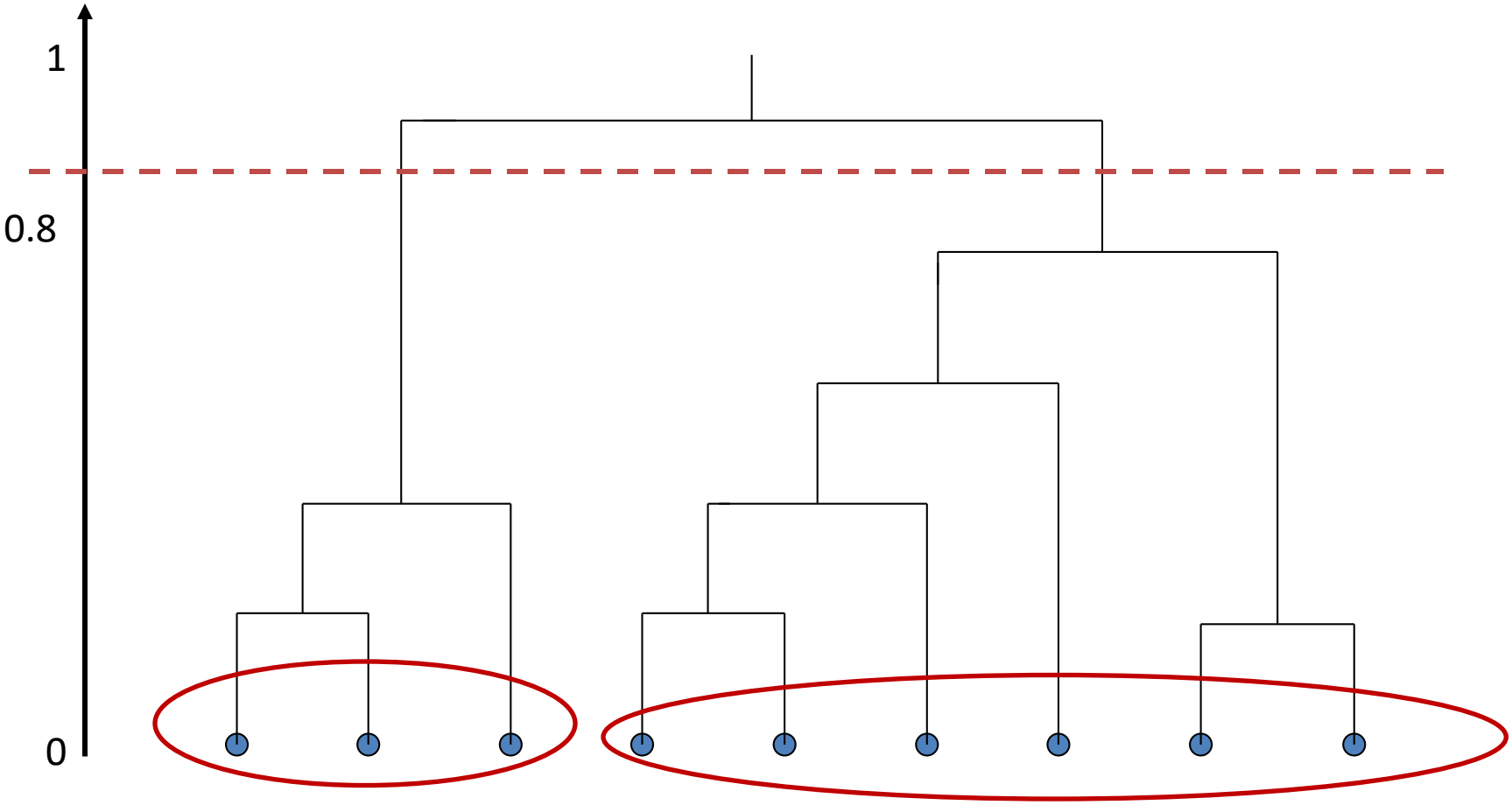
The height of connections in the dendrogram represents the distance: The higher connection, the larger distance.

Visualized Dendrograms

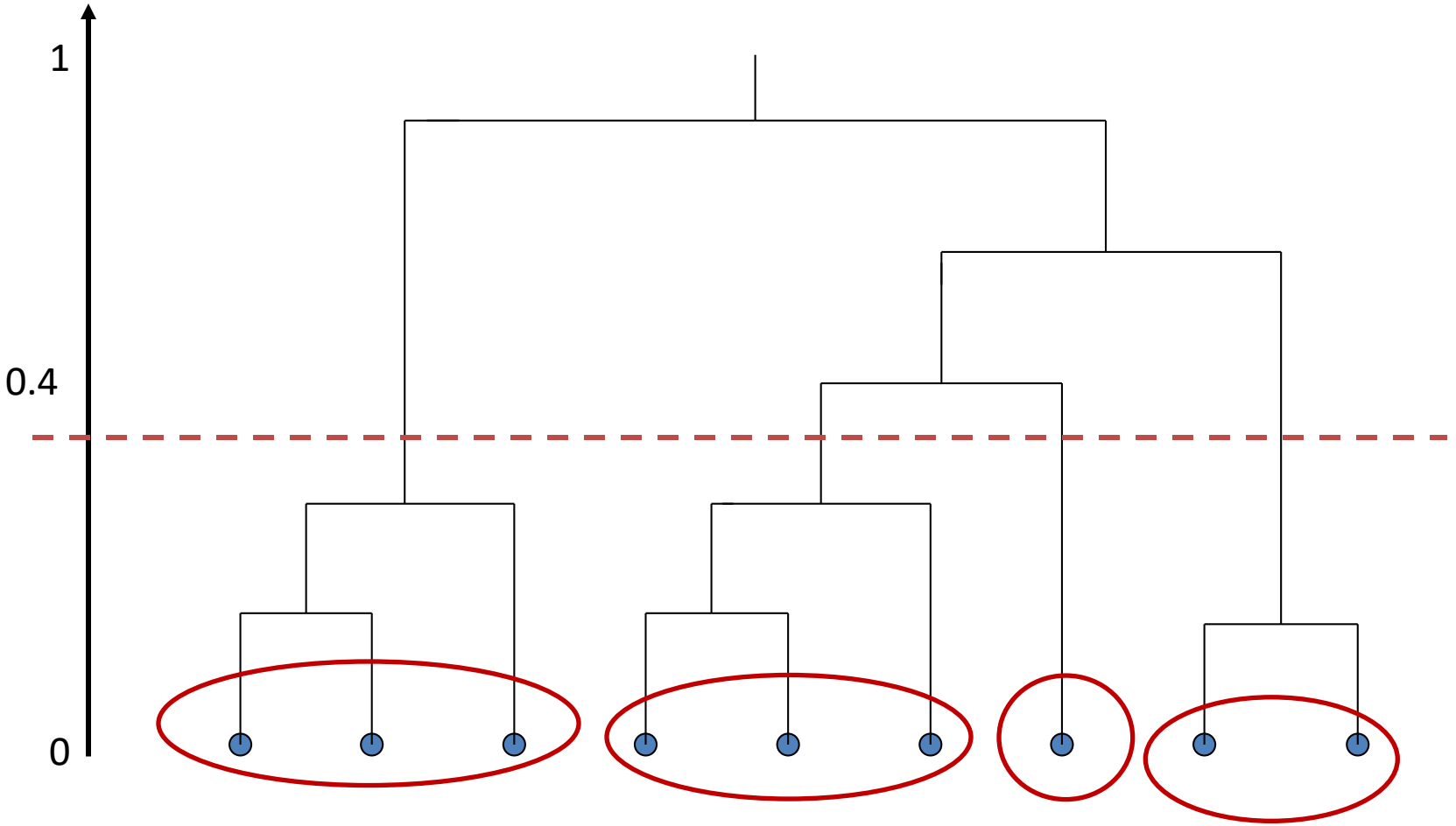


The height of connections in the dendrogram represents the distance: The higher connection, the larger distance.

Cut at D=0.8: 2 clusters



Cut at D=0.4: 4 clusters



More flexibility: You can decide number of clusters in the end

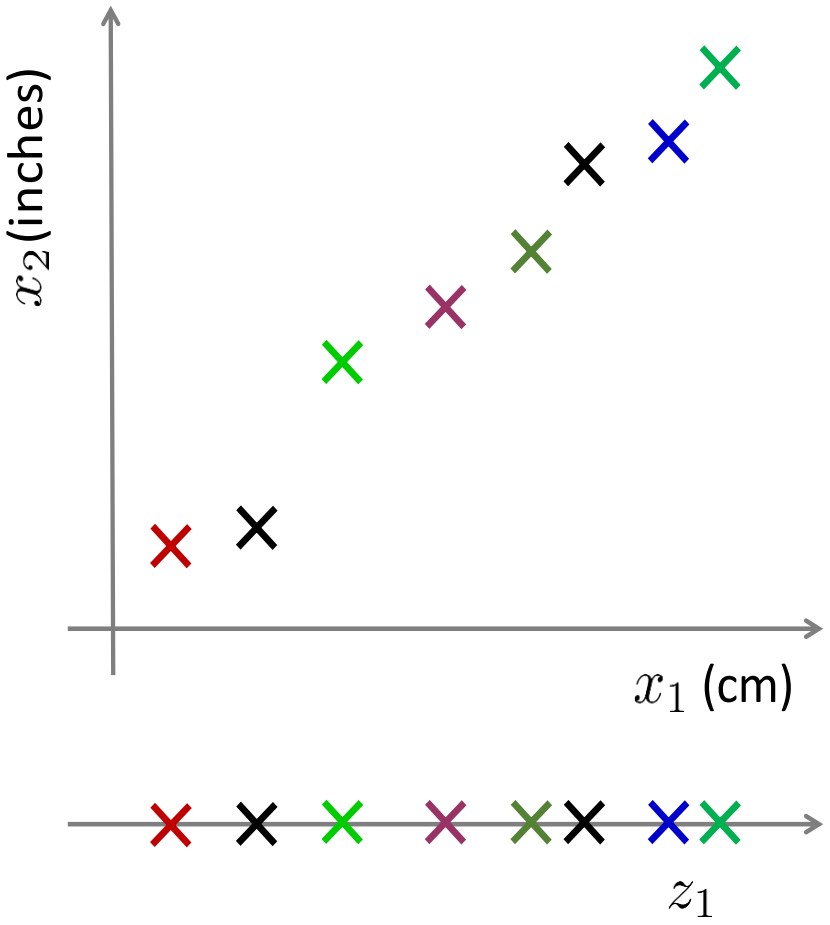
Outline

- Clustering:
  - K-means clustering
  - Dendrogram
- Dimensionality Reduction
  - Principle Component Analysis
- Ranking
  - Google's PageRank

Dimensionality Reduction

- Given data points in *d* dimensions, convert them to data points in *r*<*d* dimensions, with minimal loss of information.
- Used for statistical analysis, data compression and data visualization

Data Compression



Only projects onto x1 directly

Reduce data from 2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$

$$x^{(2)} \rightarrow z^{(2)}$$

$$\vdots$$

$$x^{(m)} \rightarrow z^{(m)}$$

Idea of Principle Component Analysis

- Reduce from n-dimension to k-dimension: Find vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

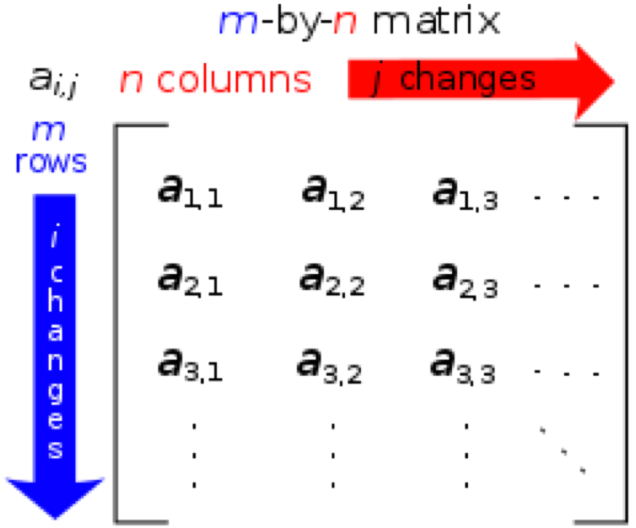
- These vectors should represent primary information of data, we call them principle components
- Begin with Linear Algebra review

Linear Algebra review

- Eigenvalue and eigenvector
- Determinant of a matrix
- Identity Matrix

Matrix

- A matrix is a rectangular array of numbers (or symbols, expressions) arranged in rows and columns
- Review by yourself: How to multiply a matrix with a vector.



Identity Matrix

- An identity matrix of size n is a n-by-n matrix with ones in the main diagonal and zeros elsewhere.
  - I.e., the entry in i-th row and j-th column of Identity matrix is 1 if i=j, is 0 otherwise
  - Use I to represent identity matrix

$$I_1 = [1],\ I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \cdots,\ I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

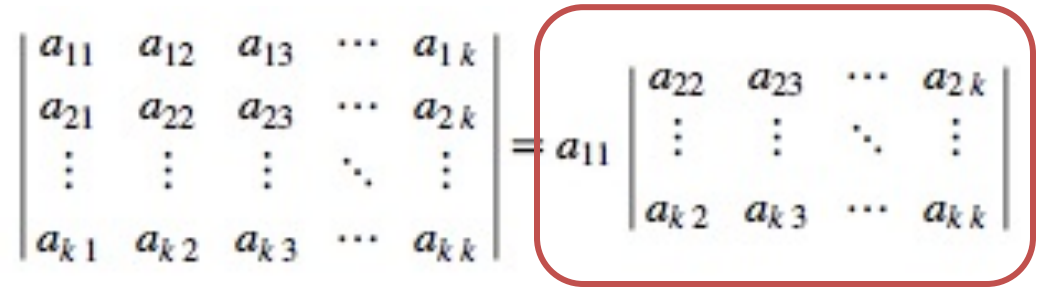
Determinant

- $|A|$ or $\det(A)$ denotes the determinant of a square matrix A
- To be simple, the determinant for 2-by-2 matrix

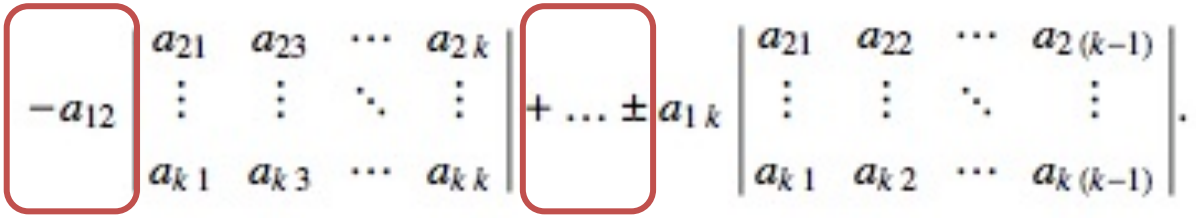
$$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

Determinant

- The determinant of a k-by-k matrix can be obtained by:



Multiply the n-th element in the first row with the sub-matrix excluding n-th column and first row



Alternate between plus and minus on subsequent entries; the first entry use “+”

Review: Eigenvalues & Eigenvectors

$$Ax = \lambda x$$

A: Square Matirx

x : Eigenvector or characteristic vector

λ : Eigenvalue or characteristic value

- ***The zero vector can not be an eigenvector***
- ***The value zero can be eigenvalue***

Example

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

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

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

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

Finding eigenvalues

- Calculate $\det(\lambda I - \mathrm{A}) = 0$
- Example: What's the eigenvalues of $\mathrm{A} = \begin{pmatrix} 2 & 9 \\ 1 & 2 \end{pmatrix}$?
  - $det \begin{pmatrix} \lambda - 2 & -9 \\ -1 & \lambda - 2 \end{pmatrix} = 0$;
  - $(\lambda - 2)(\lambda - 2) - 9 = (\lambda - 5)(\lambda + 1) = 0$;
  - λ = -1 or 5: 2 eigenvalues
- Eigenvectors can be found by solving Ax = λx
  - May not be unique: direction of vector fixed, magnitude of vector not fixed.

Review: Eigenvectors

Example 1: Find the eigenvalues of $A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$

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

$$= \lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2) = 0$$

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.

Covariance

- Covariance of two features i,j in dataset X is:
  $Cov(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$
- Here μ denotes average value, $\mathbb{E}$ denotes expectation, X_i denotes the values of i-th feature of X.
- Used to find relationships between dimensions (features) in high dimensional data sets
- When j=k, it represents variance

Covariance

- Positive covariance: both dimensions increase or decrease together, e.g. as the number of hours studied increases, the marks in that subject increase
- Negative value: while one increases the other decreases, or vice-versa
- If covariance is zero: the two dimensions are independent of each other

Covariance Matrix

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1 - \mu_1)(X_1 - \mu_1)] & \mathrm{E}[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_1 - \mu_1)(X_n - \mu_n)] \\ \mathrm{E}[(X_2 - \mu_2)(X_1 - \mu_1)] & \mathrm{E}[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n - \mu_n)(X_1 - \mu_1)] & \mathrm{E}[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & \mathrm{E}[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}$$

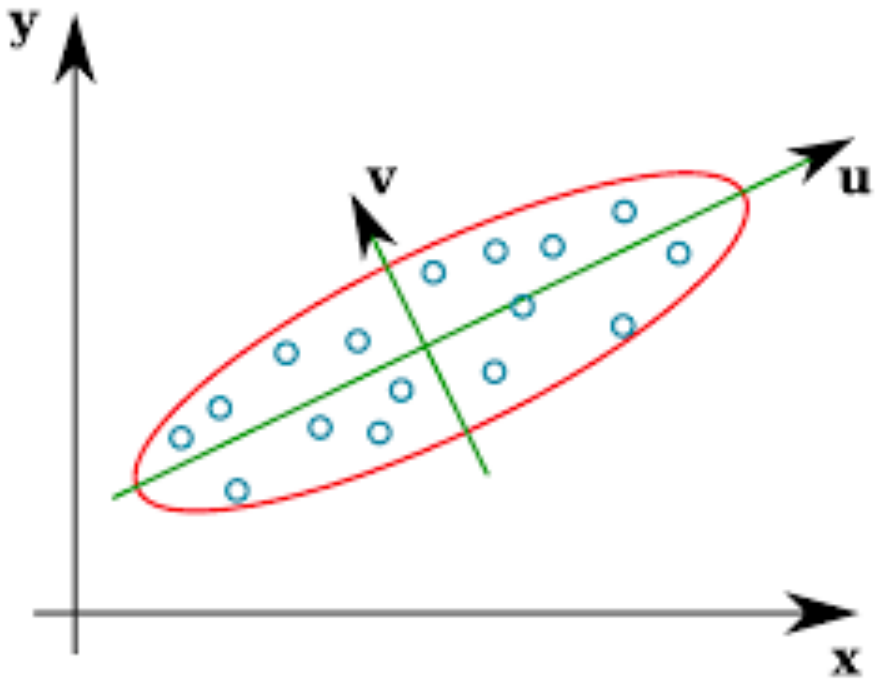
Element in the i, j position is the covariance between the i-th and j-th elements of a multi-dimension random variable.

Covariance matrix is symmetric, then its eigenvalues will be positive (not discussed today).

Intuitive Idea of PCA

- What we DON'T want for projection:
  - Original data has large variance, but projected data has small variance.
  - It means original data is spread but projected data is not: A lot of information loss during projection.
- PCA's goal: maximize the variance of projected data
  - We will show a proof in today's tutorial that maximizing the variance of projected data is equivalent to projecting to the eigenvectors corresponding to largest eigenvalues.

Visualized Explain



To keep original information of how the data is distributed, the data should be projected to the direction that maximizes the variation.

Suppose we project 2D data (x,y) to 1D, should we project to u or v?

We should project to u, projecting data to u results in a larger variance than to v.

Principle Component Analysis

- We use eigenvectors of covariance matrix as principle components (PC), the order of PCs follows the magnitude of eigenvalues. E.g., the first PC is the eigenvector corresponding to largest eigenvalue.
- In last page's example, u,v are the only two eigenvectors of covariance matrix, the length of arrow denotes their eigenvalue.
- Then 1^{st} PC is u, 2^{nd} PC is v

Principal Component Analysis

Goal: Find r-dim projection that best preserves variance

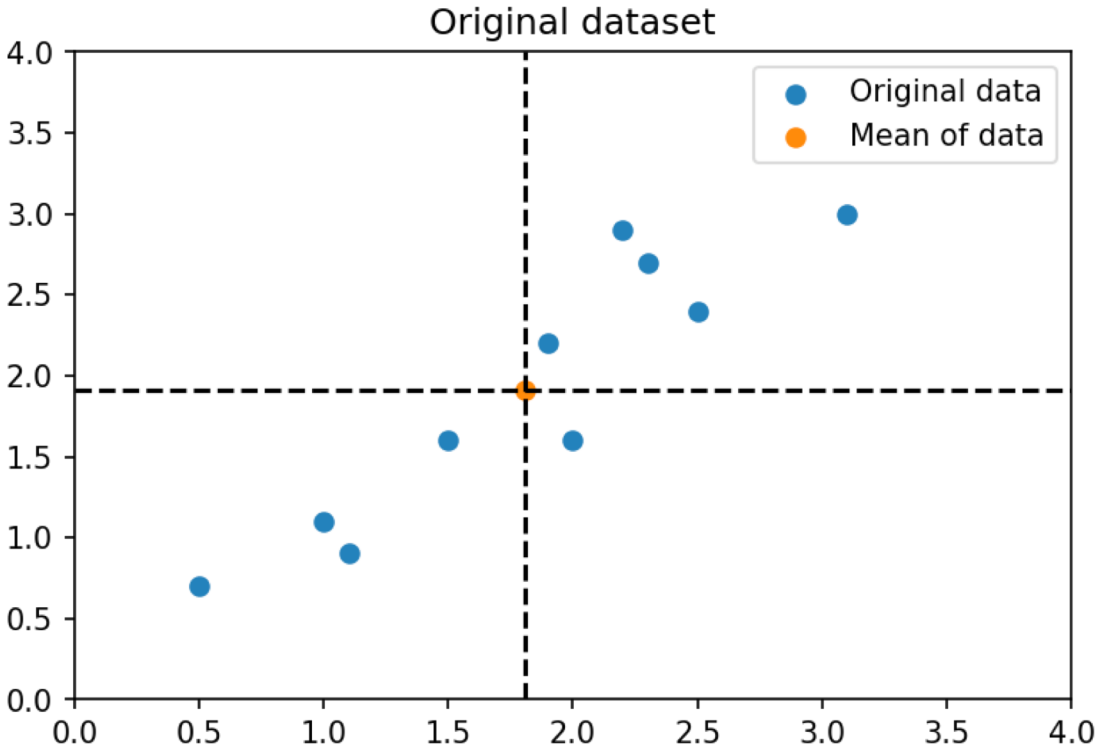
1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

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

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

More demo in detail in today's tutorial.

PCA 2D Example



Compute covariance matrix:
0.61655556,0.61544444
0.61544444,0.71655556

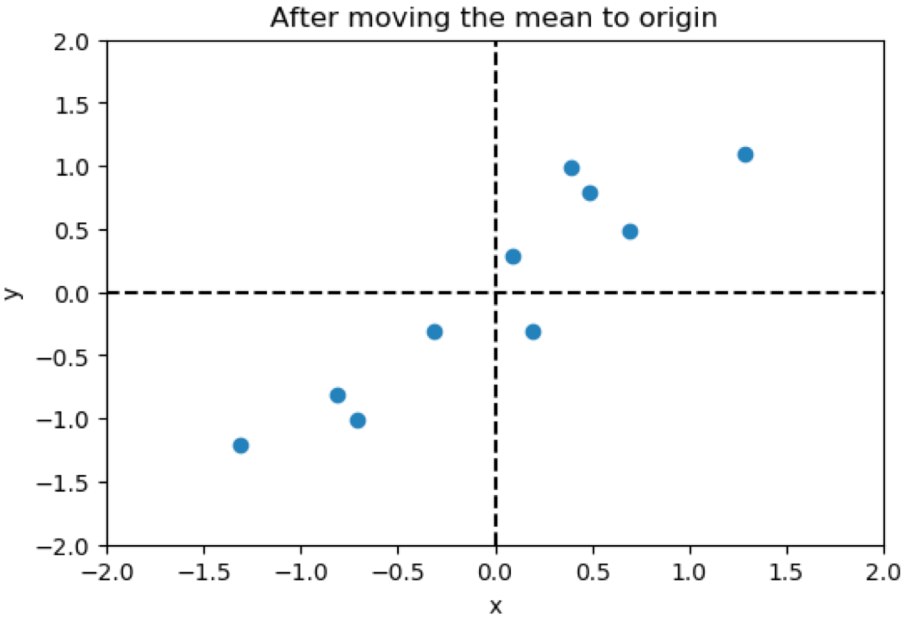
Eigenvalues:
0.0490834 ; 1.28402771

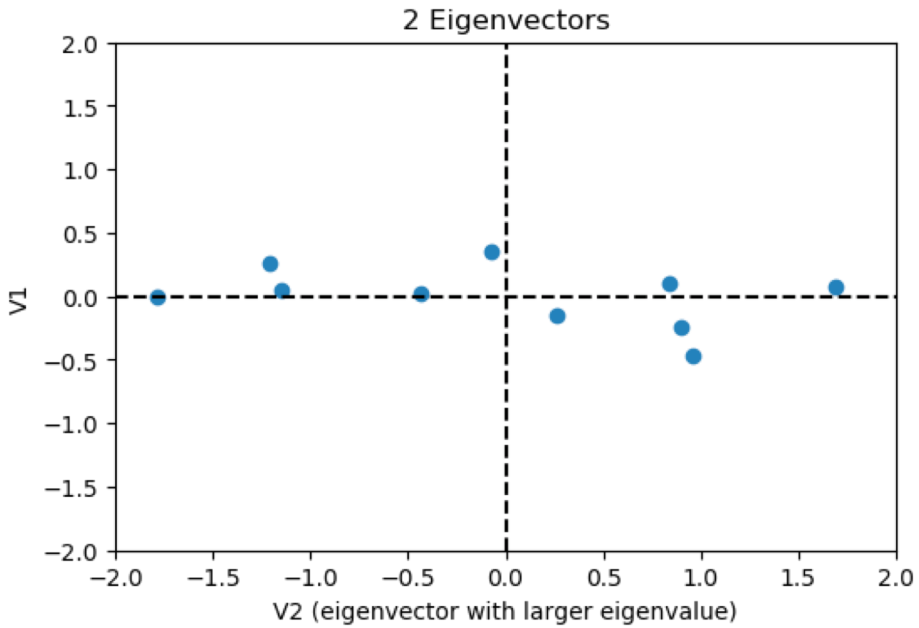
Corresponding to eigenvectors:
V1 = 0.73518 -0.67787
V2 = -0.67787 -0.73518

x = [2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1], mean=1.81

y = [2.4, 0.7, 2.9, 2.2, 3, 2.7, 1.6, 1.1, 1.6, 0.9], mean=1.91

Use 2 Eigenvectors



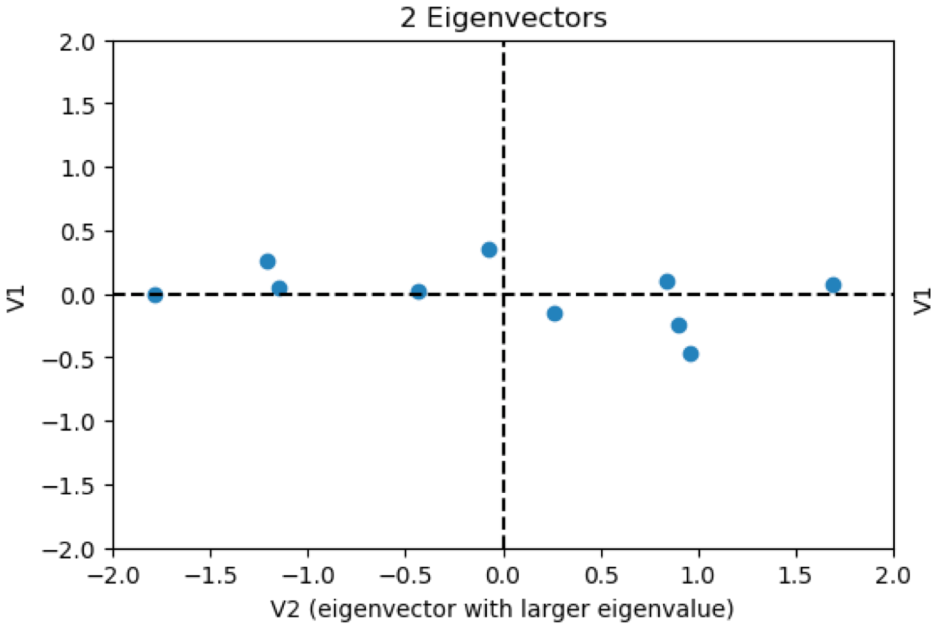


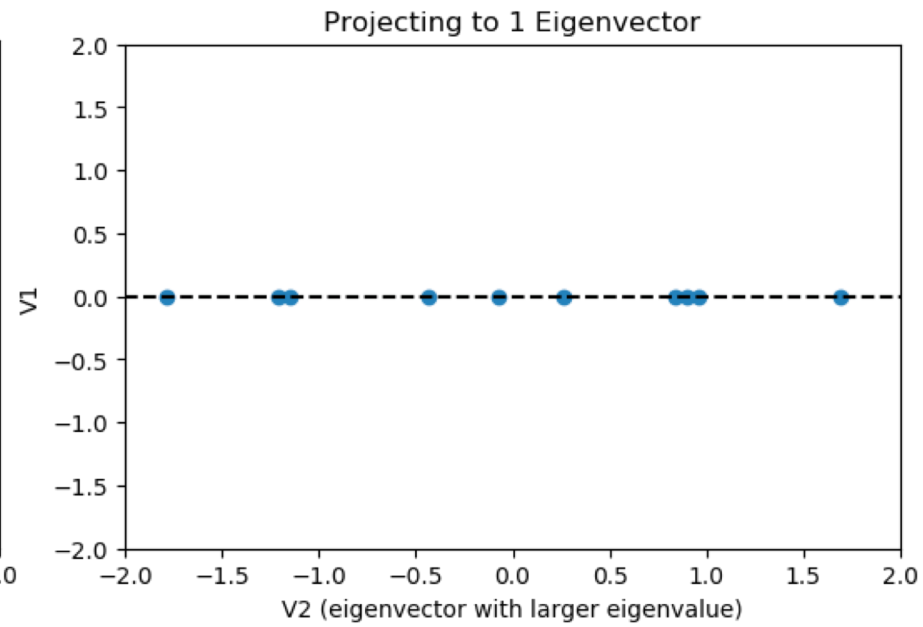
Recall the equation:

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

First minus average value to move to origin, then rotates by A. The new coordinates are V1 and V2 respectively

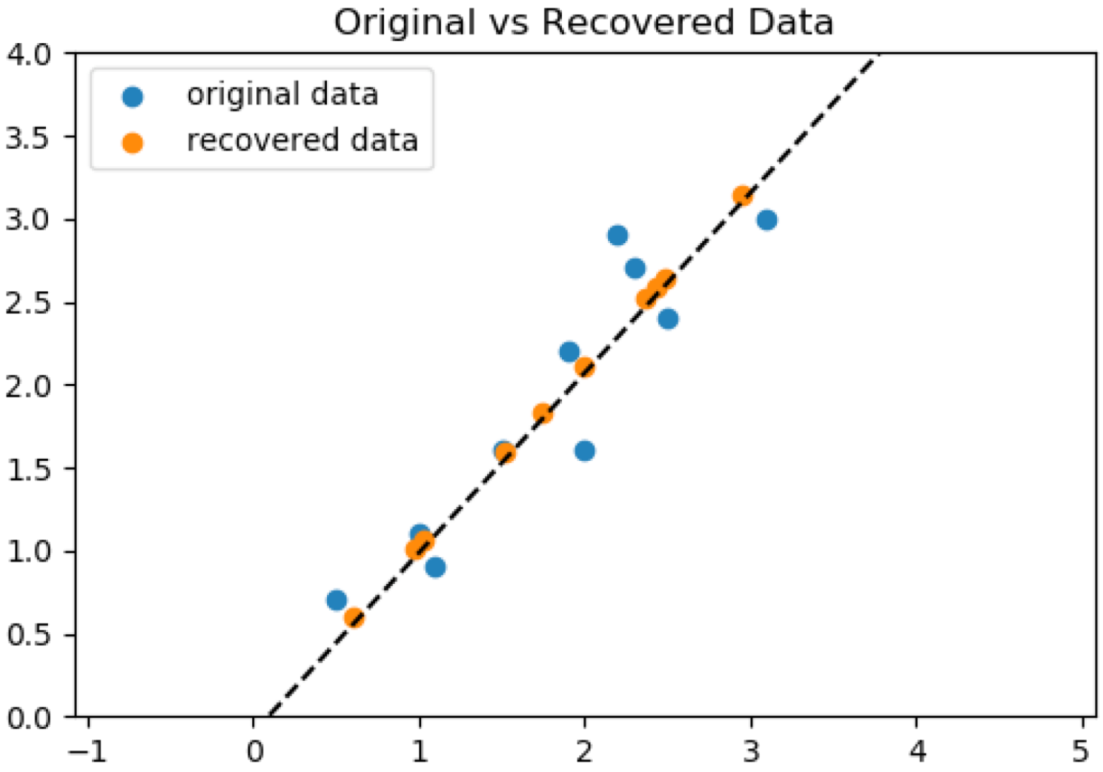
2 vs 1 Eigenvector(s)





If we use only 1 eigenvector, then PCA projects 2D data into 1D, therefore achieves data compression

PCA Has Small Information Loss



We can recover the data from 1D to 2D. The projected data (orange dots on the dashed line) is close to original data

Application: Image Compression

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

60 principle components:
Close to original image

PCA compression: 144D → 3D

3 principle components: Still represents important information of the original image.

How Many Principle Components?

- Selected principle components should explain most of the information (variance) of data.
- Define proportion of variance (PoV):
  - $PoV = \frac{\lambda_1+\lambda_2+ \ldots+\lambda_k}{\lambda_1+\lambda_2+ \ldots+\lambda_k+ \ldots+\lambda_N}$, N is total number of PC, k is number of PC selected.
- Large PoV (Close to 1): First PCs have represented a large fraction of information of data.
- Add PoV from 1 to k. Stop when PoV > 0.9

Outline

- Clustering:
  - K-means clustering
  - Hierarchical clustering
- Dimensionality Reduction
  - Principle Component Analysis
- Rating Webpages' importance
  - Google's PageRank

Background

- Why is Page Importance Rating important?
  - New challenges for information retrieval on the World Wide Web due to huge number of web pages: 150 million by1998, 1000 billion by 2008
  - Diversity of web pages: different topics, quality, etc.
  - Need to find out which pages are important first, can't go through all of them.
- What is PageRank? A method for rating the importance of web pages using the link structure of the web.

The History of PageRank

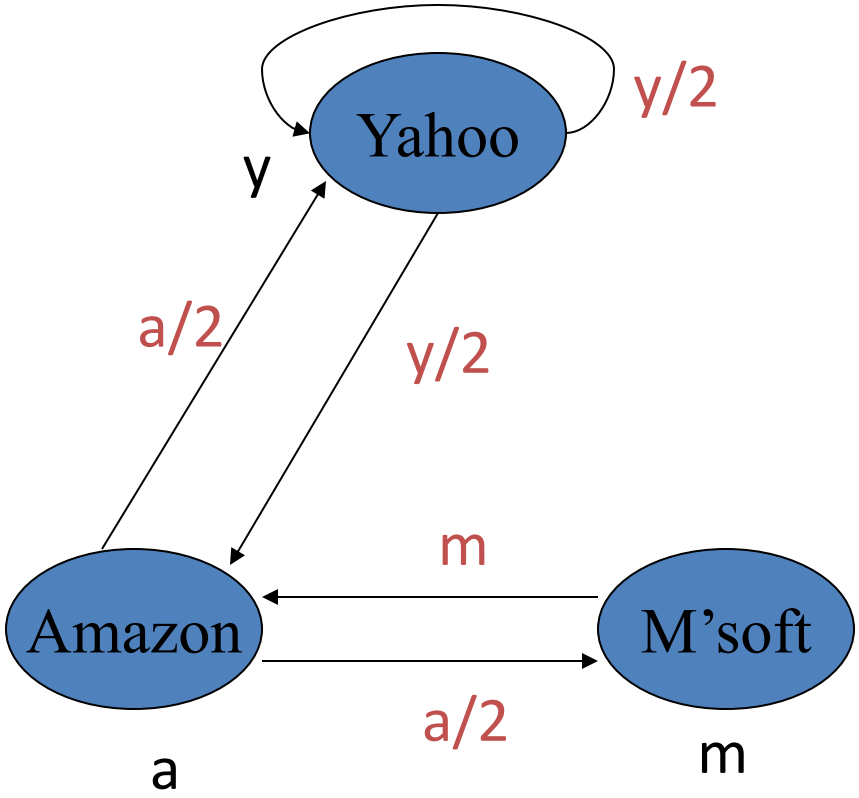
- PageRank was developed by Larry Page (hence the name *Page*-Rank) and Sergey Brin.
- It is first as part of a research project about a new kind of search engine. That project started in 1995 and led to a functional prototype in 1998.
- Shortly after, Page and Brin founded Google ($$$$).

- From 2016, PageRank is removed from Google toolbar but still used in Google internally.

Simple Recursive Formulation

- Each link's vote is proportional to the importance of its source page
- If page P with importance x has n out-links, each link gets x/n votes
- Page P's own importance is the sum of the votes on its in-links
- Final PageRank score: Importance=sum of votes from all in-links

Simple Recursive Model



$$y = y/2 + a/2$$
$$a = y/2 + m$$
$$m = a/2$$

- Each link's vote is proportional to the importance of its source page
- If page P with importance x has n outlinks, each link gets x/n votes
- Page P's own importance is the sum of the votes on its inlinks

Solving the Equations

- 3 equations, 3 unknowns, no constants
  - No unique solution
  - All solutions equivalent modulo scale factor
- Additional constraint forces uniqueness
  - y+a+m = 1
  - y = 2/5, a = 2/5, m = 1/5

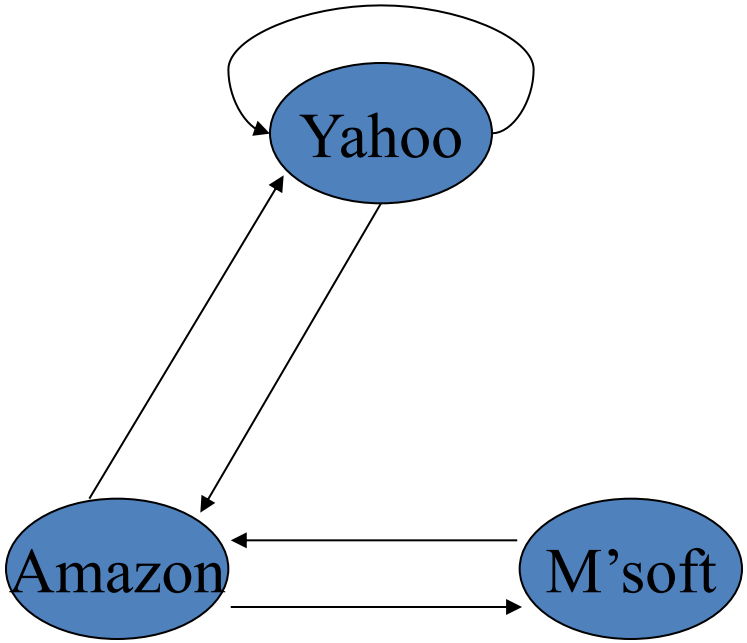
Random Walk Interpretation

- An equivalent view of PageRank.
- Imagine a random web surfer
  - At any time t, surfer is on some page P
  - At time t+1, the surfer follows an outlink from P uniformly at random.
  - Ends up on some page Q linked from P, Process repeats indefinitely
- Let **p**(t) be a vector whose i^{th} component is the probability that the surfer is at page i at time t
  - **p**(t) is a probability distribution on pages

Stationary Distribution

- Where is the surfer at time t+1?
  - Follows a link uniformly at some probability
  - **p**(t+1) = **Mp**(t) where M is the transition probability
- Suppose the random walk reaches a state such that **p**(t+1) = **Mp**(t) = **p**(t)
  - Then **p**(t) is called a stationary distribution for the random walk
  - The PageRank score r is the stationary distribution, can be solved by r=Mr.
  - Normalization by scaling sum of r to 1.

The Probability Matrix



Pr{M'soft to Amazon}=1

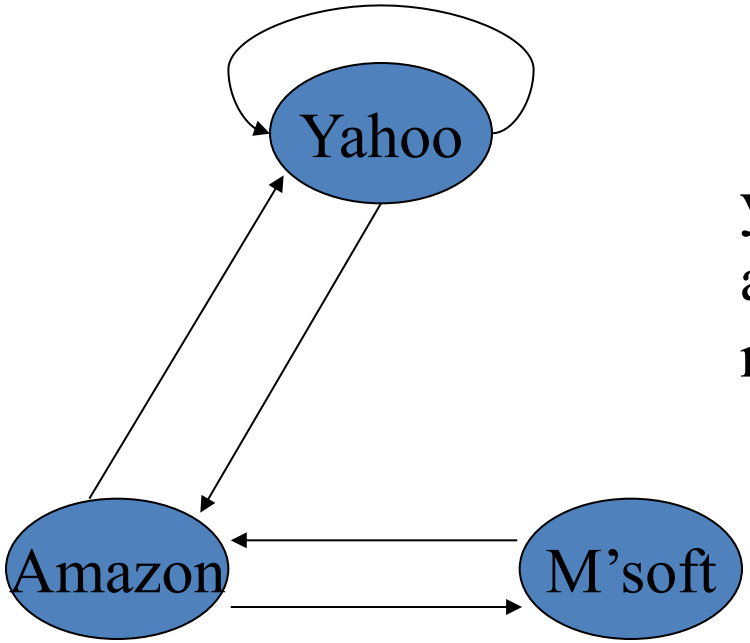
	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

Entry on I-th row J-th column: Probability to walk from node J to I.

Stationary Distribution=PageRank Score

- Stationary distribution represents PageRank score.
  - PageRank Score: A node's importance equals to the vote from adjacent nodes.
  - Stationary Distribution: Probability of being at one node equals to sum of probability coming from other nodes
  - Both of them describes the stable state.

Stationary Distribution=PageRank Score



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

Flow model:

y = y /2 + a /2
a = y /2 + m
m = a /2

Equivalent

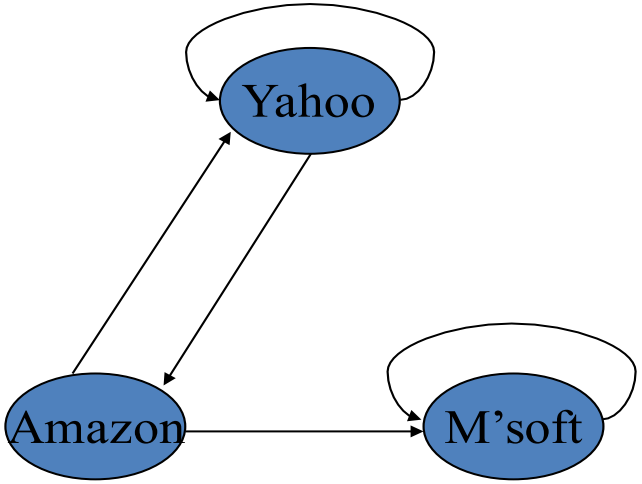
Stationary Distribution:

$(y, a, m)^T = M \times (y, a, m)^T$

y = y /2 + a /2 + 0 m
a = y /2 + 0 a + 1 m
m = 0 y + a /2 + 0 m

Practice Question

- Can you calculate the PageRank score r? Note the graph is different from previous page.



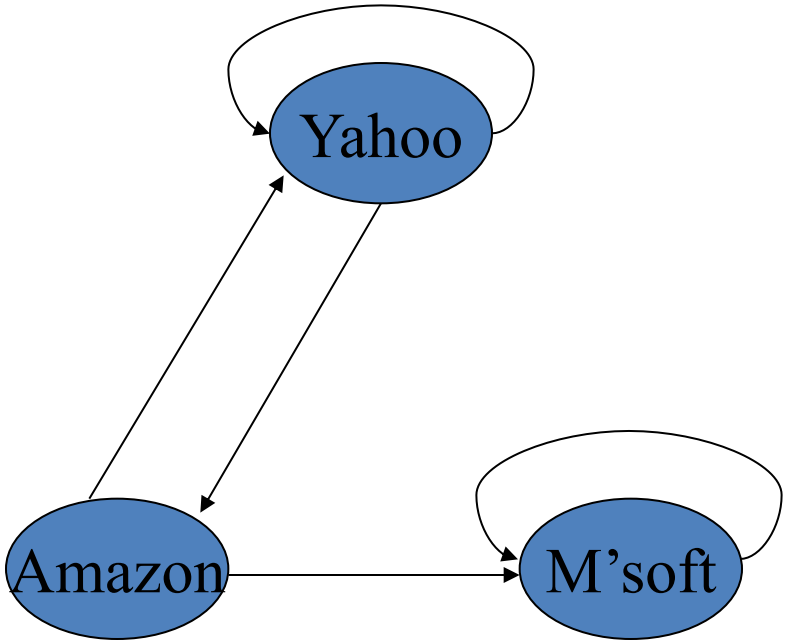
	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

- Answer: It's (0,0,1)
  - Microsoft is very important, other two pages are useless!
  - How can this happen?

Spider Traps

- A group of pages is a spider trap if there are no links from within the group to outside the group
  - Random surfer gets trapped, it continuously walk in the trap.
- Spider traps violate the conditions needed for the random walk theorem

M’soft Becomes a Spider Trap



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

The result of M’soft will be 1 in the end: Yahoo & Amazon has some probability to walk to Microsoft, but Microsoft never walks out

Avoid Traps: Damping Factor

- PR(A) = (1-d) + d(PR(T1)/C(T1) +…+PR(Tn)/C(Tn))

Original PageRank score

- **PR(Tn)** - Each page has a notion of its own self-importance. That's "PR(T1)" for the first page in the web all the way up to "PR(Tn)" for the last page

- **C(Tn)** - Each page spreads its vote out evenly amongst all of it's outgoing links. The count, or number, of outgoing links for page 1 is "C(T1)", "C(Tn)" for page n, and so on for all pages.

Avoid Traps: Damping Factor

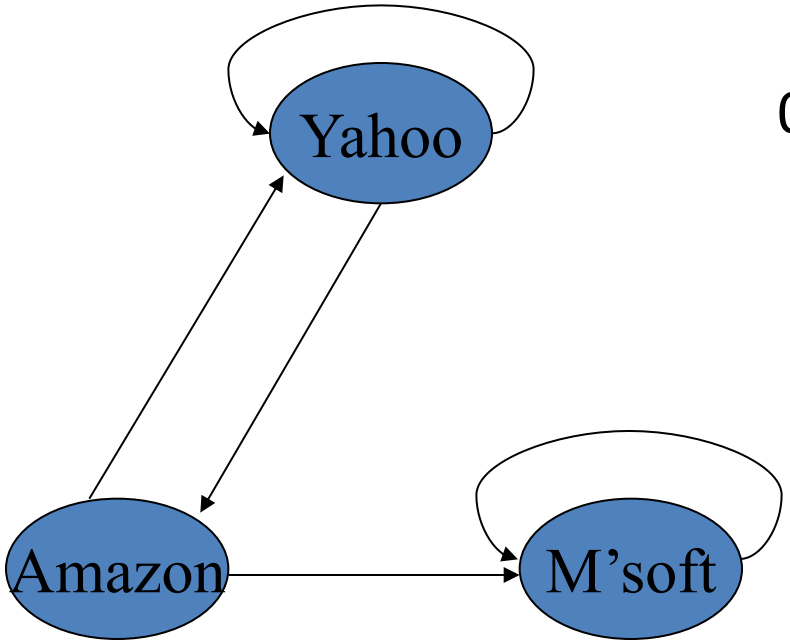
- **d(…** - All these fractions of votes are added together but, to stop the other pages having too much influence, this total vote is “damped down” by multiplying it by the factor “d”

- The result should be normalized (scale to sum of PR=1)

Equivalent View of Damping Factor

- **d(…** - total vote is “damped down” by multiplying it by the factor “d”
- With a probability (1-d), restart the random surfing at any position uniformly at random.
- Avoids stuck in spider traps.

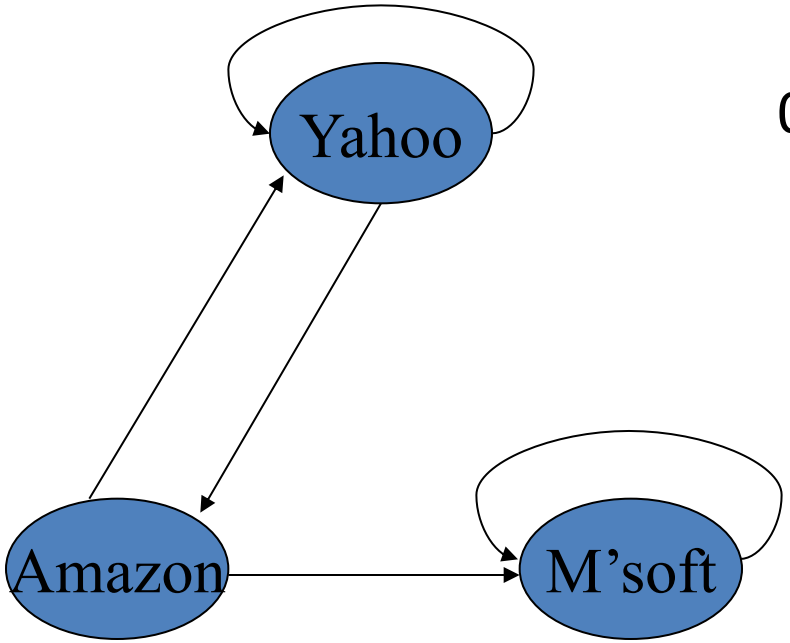
Equivalent View (d = 0.8)



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

Practice Question (d=0.8)



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$= \begin{array}{c} \\ y \\ a \\ m \end{array} \begin{array}{c} \begin{array}{ccc} y & a & m \end{array} \\ \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix} \end{array}$$

Can you find the Page Rank for these pages?

PR(Yahoo) = 7/33, PR(A)=5/33, PR(M)=7/11,
Remember to scale sum of PR to 1.

Wrap-up

- Unsupervised learning: Learning without labels.
- Clustering:
  - K-means clustering
  - Hierarchical clustering
- Dimensionality Reduction
  - Principle Component Analysis
- Rating Webpages' importance
  - Google's PageRank