Simulations in Statistical Physics

Course for MSc physics students

Janos Török

Department of Theoretical Physics

May 11, 2023

Clustering, modularity, community detection

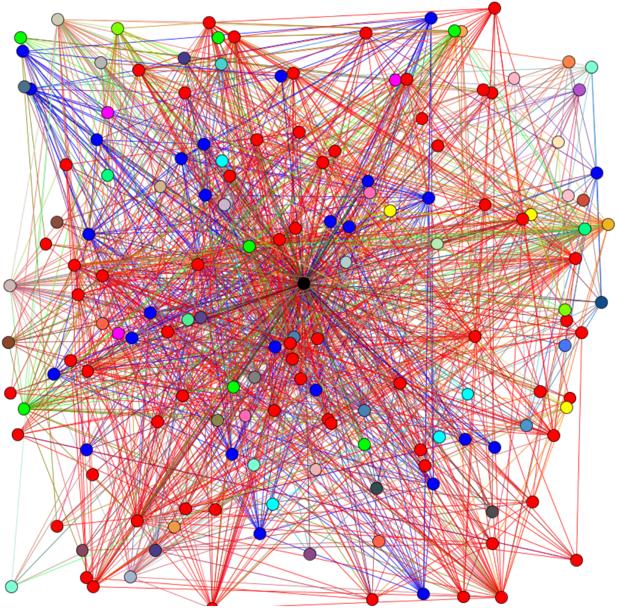
Patterns in comlpex network

- Natural networks are not homogeneous
- There are natural groups
- These groups are more densely connected internally then externally
- Nodes in groups are more similar
- Exact mathematical definition is lacking
- These groups are called *communities*
- *Clustering*: group similar items together

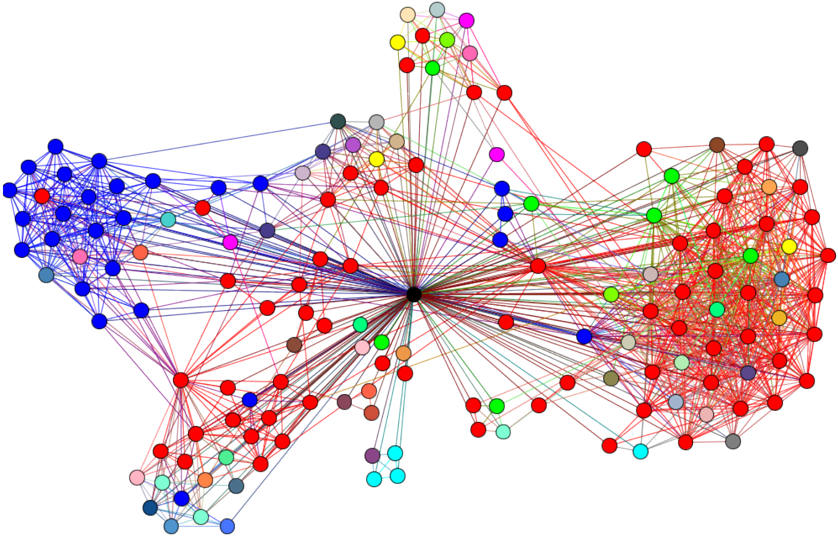
Communities/Clustering

- Networks: Communities
  - More connected to itself than to the rest
  - No unique definition
  - Mesoscopic structures
- Any data (incl. networks): Clusters
  - Set of data points more similar to eachother then to the rest

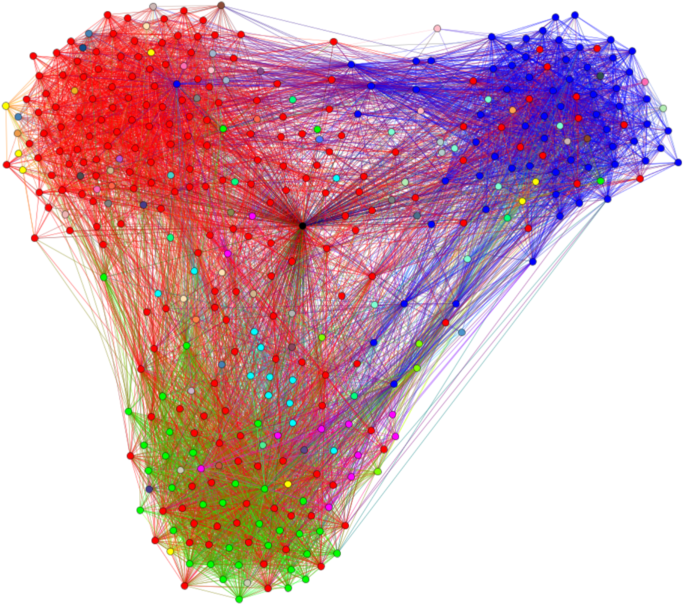
Egocentric network on iwiw



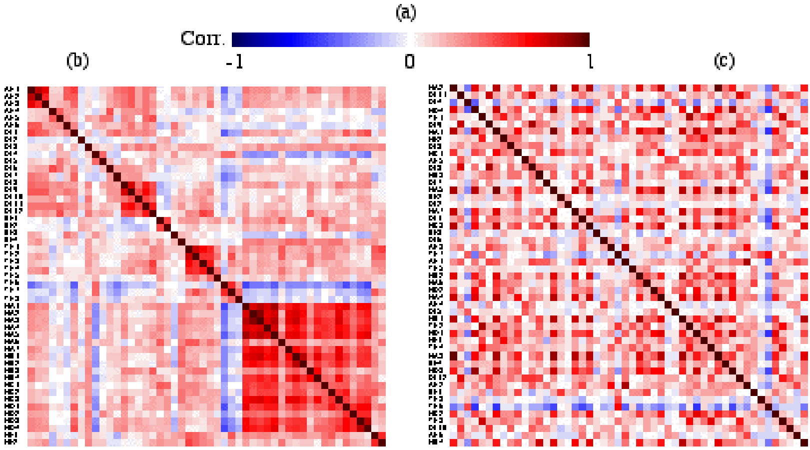
Egocentric network on iwiw



Egocentric network on iwiw



Clustering example: Correlation between 50 symptoms

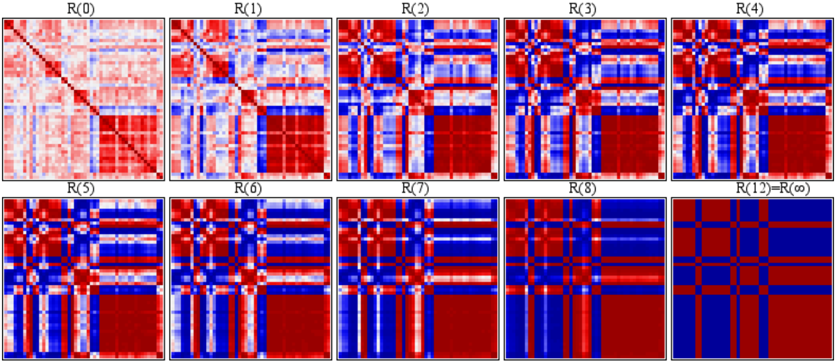


Clustered

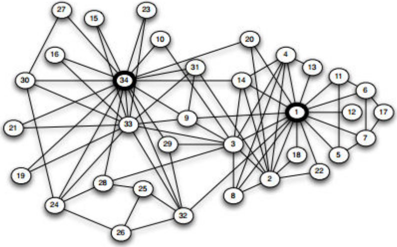
Random

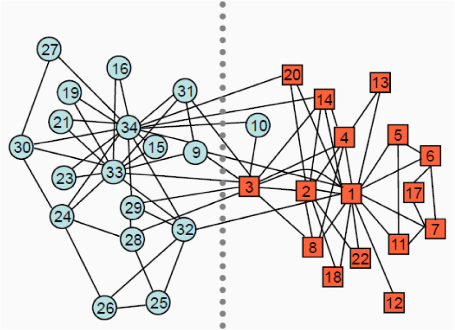
Clustering example: Correlation between 50 symptoms

Community detection



Zachary karate club



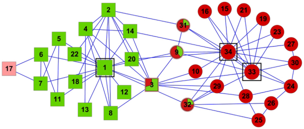


Cluster, Community definition:

- Group which is more connected to itself than to the rest
- Group of items which are more similar to each other than to the rest of the system.

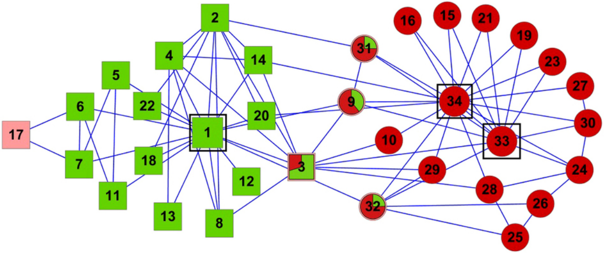
Communities, Partioning:

- Strict partitioning clustering: each object belongs to exactly one cluster
- Overlapping clustering: each object may belong to more clusters
- Hierarchical clustering: objects that belong to a child cluster also belong to the parent cluster
- Outliers: which do not conform to an expected pattern



Communities, Partitioning

- Strict partitioning clustering: each object belongs to exactly one cluster
- Overlapping clustering: each object may belong to more clusters
- Hierarchical clustering: objects that belong to a child cluster also belong to the parent cluster
- Outliers: which do not conform to an expected pattern



Communities, Partitioning, definitions:

- Local:
  - (Strong) Each node has more neighbors inside than outside
  - (Weak) Total degree within the community is larger than the total degree out of it.
  - Modularity by local definition (above)
  - Clique-percolation
- Global: The community structure found is optimal in a global sense
  - Modularity
  - k-means clustering
  - Agglomerative hierarchical clustering

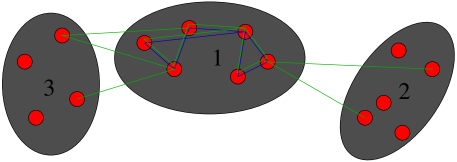
Communities, Partitioning, definitions:

- Hundreds of different algorithms, definitions
- Starting point: *adjacency matrix* A_{ij}, the strength of the link between nodes i and j
- Nodes as vectors (e.g. rows of adjacency matrix)
- Metric between nodes: $||a - b||$:
  - Euclidean distance: $||a - b||_2 = \sqrt{\sum_i (a_i - b_i)^2}$
  - Maximum distance: $||a - b||_\infty = \max_i |a_i - b_i|$
  - Cosine similarity: $||a - b||_c = \frac{a \cdot b}{||a|| \, ||b||}$
  - Hamming distance: number of different coordinates

Modularity

Global method

- e_{ii} percentage of edges in module (cluster) i
  probability edge is in module i
- a_i percentage of edges with at least 1 end in module i
  probability a random edge would fall into module i



- Modularity is

$$Q = \sum_{i=1}^{k} (e_{ii} - a_i^2)$$

- Try to maximize Q

Modularity algorithm

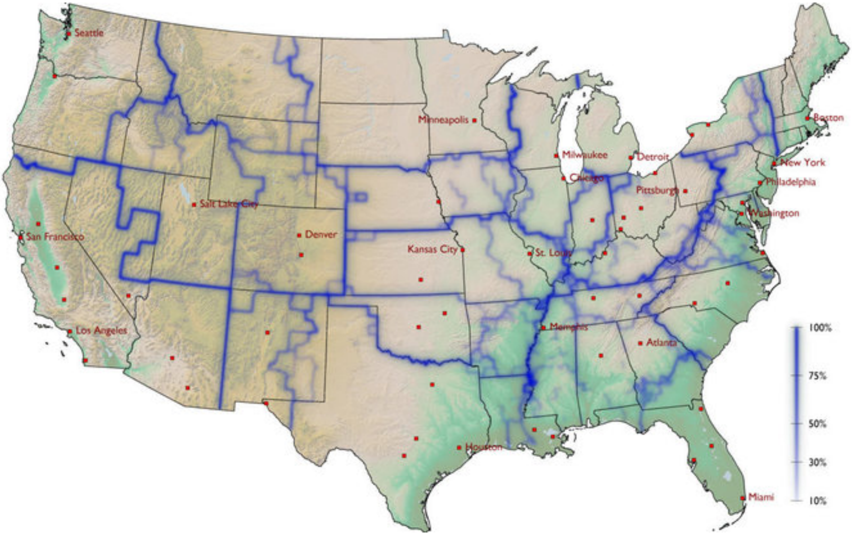
- Rewrite Q:
$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right]$$
$2m = \sum_i k_i$
- Only two modules
- $s_i = \pm 1$: 1 if node i is in module 1; -1 otherwise
$$Q = \frac{1}{4m} \sum_{\{i,j\}} \left[A_{ij} - \frac{k_i k_j}{2m} \right] (s_i s_j + 1)$$
- $+1$ is a constat can be omitted
- Change the vector s_i to maximize Q

Modularity algorithm

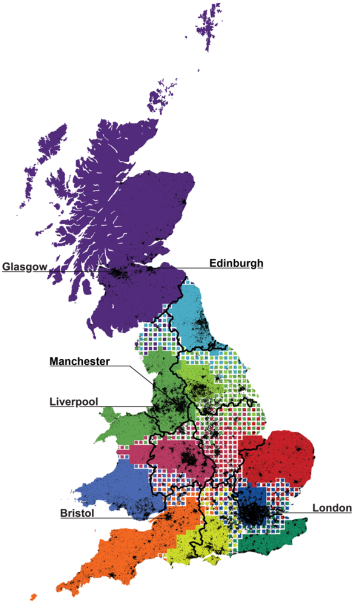
$$Q = \frac{1}{4m} \sum_{\{i,j\}} \left[A_{ij} - \frac{k_i k_j}{2m} \right] s_i s_j$$

- Try to find ± 1 vector s_i that maximizes the modularity.
- Start with two groups
- Then split one of the two groups using the same technique
- Very similar to spin glass Hamiltonian
- Generally a np-complete problem, we can use the same techniques.
- Often steepest descent is used, (greedy method): change the site that would increase the modularity the most.

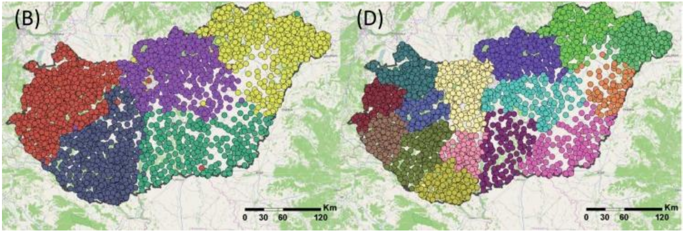
Modularity: human interactions between cities



Modularity: human interactions between cities



iWiW vs. counties: aggregate connections between cities



Problems with modularity

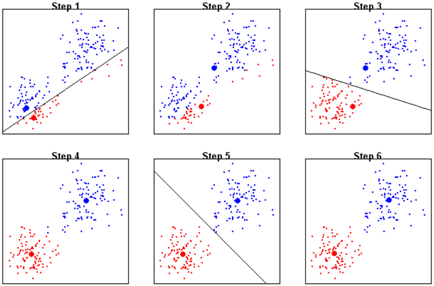
Resolution

$$Q = \frac{1}{4m} \sum_{\{i,j\}} \left[A_{ij} - \frac{k_i k_j}{2m} \right] s_i s_j$$

- On large networks normalization factor m can be very large
- (It relies on random network model)
- The expected edge between modules decreases and drops below 1
- A single link is a strong connection.
- Small modules will not be found

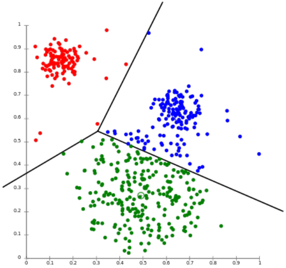
k-means clustering

- Cut the system into exactly k parts
- Let μ_i be the mean of each cluster (using a metric)
- The cluster i is the set of points which are closer to μ_i than to any other μ_j
- The result is a partitioning of the data space into Voronoi cells



k-means clustering, standard algorithm:

- Define a norm between nodes
- Give initial positions of the means m_i
- Assignment step: Assign each node to cluster whoose mean m_i is the closest to node.
- Update step: Calculate the new means of the clusters
- Go to Assignment step.

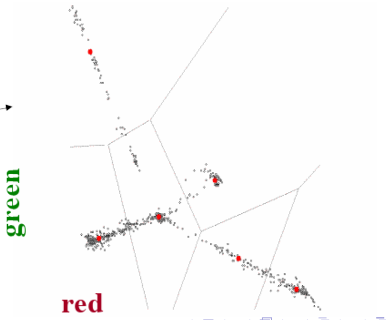


k-means clustering: Major usage

- Detection of connected parts in images
- Use the Red, Greem Blue value of each pixel
- Put them on a 3d space
- Find relevant clusters

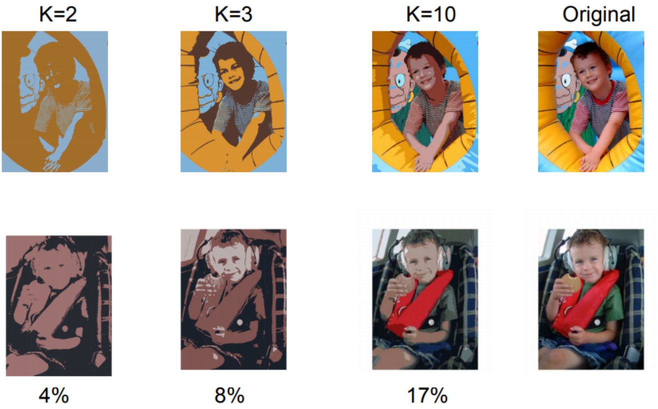
Image with 4 color clusters

Normalized color plots according to **red** and **green** components.



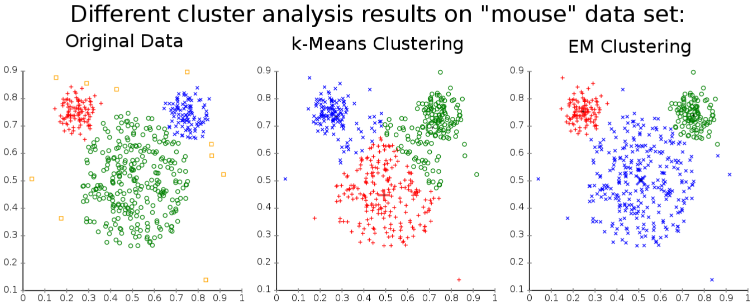
k-means clustering: Image color segmentation

- Detection of connected parts in images
- Use the Red, Greem Blue value of each pixel
- Put them on a 3d space
- Find relevant clusters
- Use the center instead of each color
- Define connected clusters as objects on image

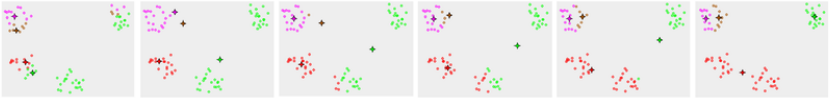


k-means clustering: Problems

- k has to fixed beforhand
- Fevorizes equal sized clusters:



- Very sensitive on initial conditions:



- No guarantee that it converges

Hierarchical clustering

1. Define a norm between nodes $d(a, b)$
2. At the beginning each node is a separate cluster
3. Merge the two closest clusters into one
4. Repeat 3.

Norm between clusters $||A - B||$

- Maximum or complete linkage clustering:

$$\max\{d(a, b) : a \in A, b \in B\}$$

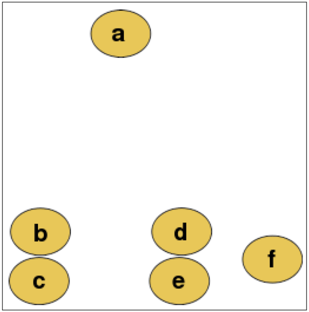
- Minimum or single-linkage clustering:

$$\min\{d(a, b) : a \in A, b \in B\}$$

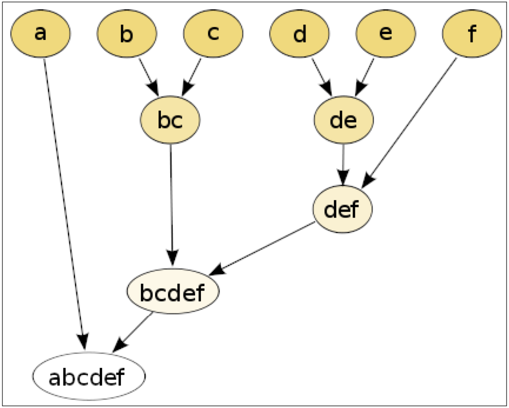
- Mean or average linkage clustering:

$$\frac{1}{||A|| \, ||B||} \sum_{a \in A} \sum_{b \in B} d(a, b)$$

Hierarchical clustering

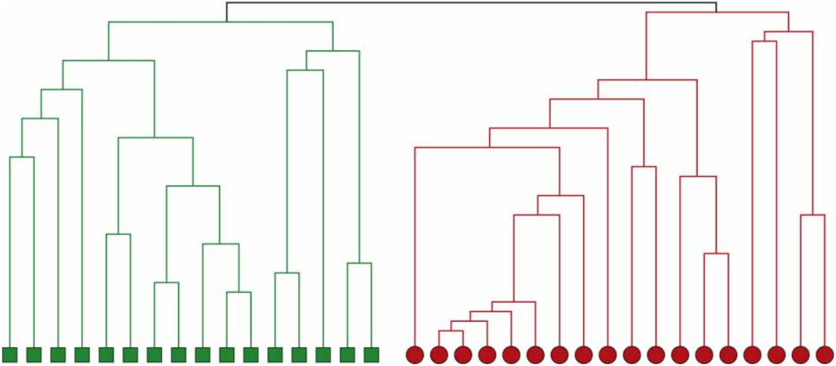


Original



Dendogram

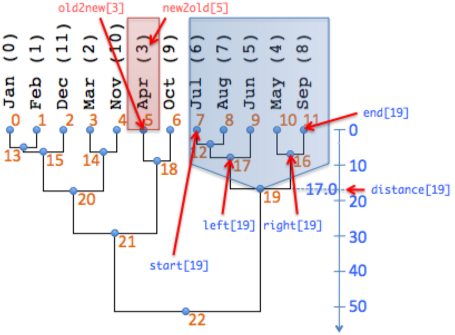
Dendograph of the Zachary karate club



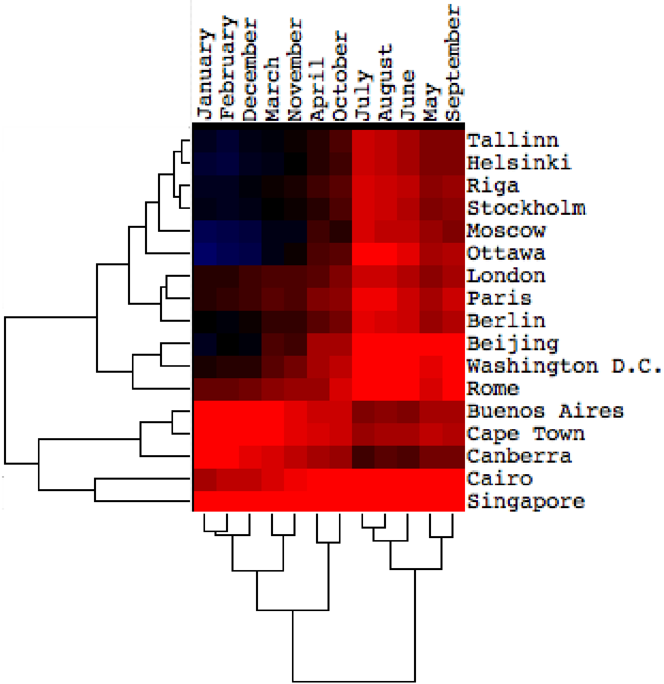
Example: Temperatures in capitals

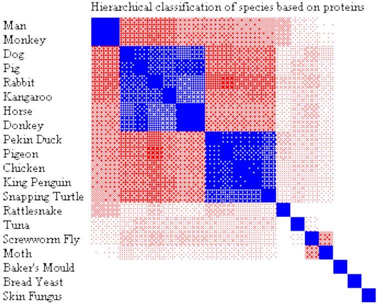
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Tallinn	-3	-5	-1	3	10	13	16	15	10	6	1	-2
Beijing	-3	0	6	13	20	24	26	25	20	13	5	-1
Berlin	0	-1	4	7	12	16	18	17	14	9	4	1
Buenos Aires	23	22	20	16	13	10	10	11	13	16	18	22
Cairo	13	15	17	21	25	27	28	27	26	23	19	15
Canberra	20	20	17	13	9	6	5	7	9	12	15	18
Cape Town	21	21	20	17	15	13	12	13	14	16	18	20
Helsinki	-5	-6	-2	3	10	13	16	15	10	5	0	-3
London	3	3	6	7	11	14	16	16	13	10	6	5
Moscow	-8	-7	-2	5	12	15	17	15	10	3	-2	-6
Ottawa	-10	-8	-2	6	13	18	21	20	14	7	1	-7
Paris	3	4	7	10	13	16	19	19	16	11	6	5
Riga	-3	-3	1	5	11	15	17	16	12	7	2	-1
Rome	8	8	11	12	17	20	23	23	21	17	12	9
Singapore	27	27	28	28	28	28	28	28	27	27	27	26
Stockholm	-2	-3	0	3	10	14	17	16	11	6	1	-2
Washington D.C.	2	3	7	13	18	23	26	25	21	15	9	3

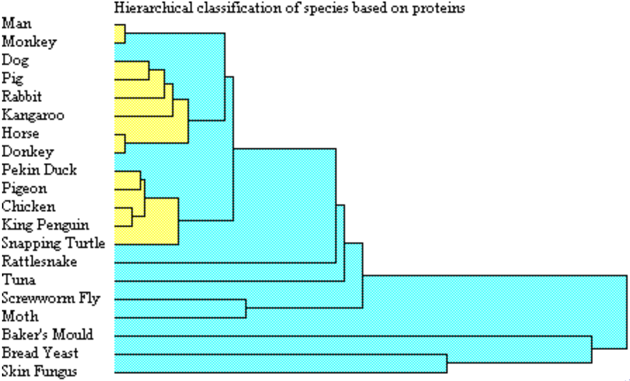
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Tallinn	-3	-5	-1	3	10	13	16	15	10	6	1	-2
Beijing	-3	0	6	13	20	24	26	25	20	13	5	-1
Berlin	0	-1	4	7	12	16	18	17	14	9	4	1
Buenos Aires	23	22	20	16	13	10	10	11	13	16	18	22
Cairo	13	15	17	21	25	27	28	27	26	23	19	15
Canberra	20	20	17	13	9	6	5	7	9	12	15	18
Cape Town	21	21	20	17	15	13	12	13	14	16	18	20
Helsinki	-5	-6	-2	3	10	13	16	15	10	5	0	-3
London	3	3	6	7	11	14	16	16	13	10	6	5
Moscow	-8	-7	-2	5	12	15	17	15	10	3	-2	-6
Ottawa	-10	-8	-2	6	13	18	21	20	14	7	1	-7
Paris	3	4	7	10	13	16	19	19	16	11	6	5
Riga	-3	-3	1	5	11	15	17	16	12	7	2	-1
Rome	8	8	11	12	17	20	23	23	21	17	12	9
Singapore	27	27	28	28	28	28	28	28	27	27	27	26
Stockholm	-2	-3	0	3	10	14	17	16	11	6	1	-2
Washington D.C.	2	3	7	13	18	23	26	25	21	15	9	3



Euclidean distance

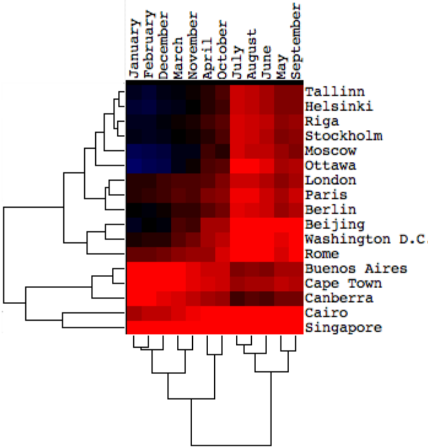

January
February
December
March
November
April
October
July
August
June
May
September
Tallinn
Helsinki
Riga
Stockholm
Moscow
Ottawa
London
Paris
Berlin
Beijing
Washington D.C.
Rome
Buenos Aires
Cape Town
Canberra
Cairo
Singapore


Hierarchical classification of species based on proteins
Man
Monkey
Dog
Pig
Rabbit
Kangaroo
Horse
Donkey
Pekin Duck
Pigeon
Chicken
King Penguin
Snapping Turtle
Rattlesnake
Tuna
Screwworm Fly
Moth
Baker's Mould
Bread Yeast
Skin Fungus


Hierarchical classification of species based on proteins
Man
Monkey
Dog
Pig
Rabbit
Kangaroo
Horse
Donkey
Pekin Duck
Pigeon
Chicken
King Penguin
Snapping Turtle
Rattlesnake
Tuna
Screwworm Fly
Moth
Baker's Mould
Bread Yeast
Skin Fungus

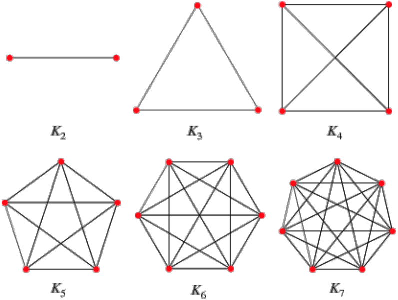
Hierarchical clustering: problems

- Advantages
  - Simple
  - Fast
  - Number of clusters can be controlled
  - Hierarchical relationship
- Disadvantages
  - No a priori cutting level
  - Meaning of clusters unclear
  - Important links may be missed
  - Different result if one item omitted



Clique percolation

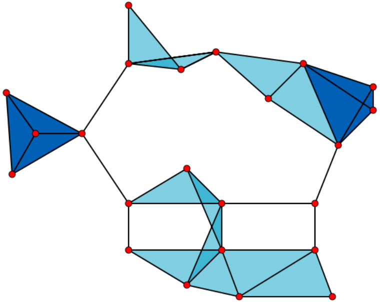
- Motication: clusters are formed with at least triangles
- Can be generalized to any k-clique

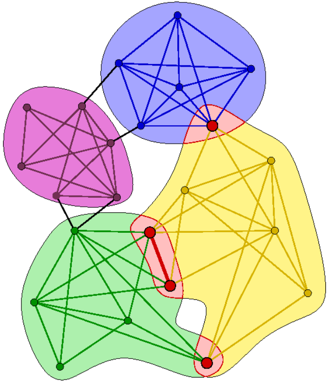


- $k = 2$ normal percolation

Clique percolation

- It will definitely lead to overlapping communities, but overlap is limited to $k - 1$ nodes
- k-clusters are included in $k - 1$ clusters





Clique percolation

- Algorithm
  - Similar to normal percolation on networks but with multiple loops
- Advantages
  - Different level of clusters
  - Clusters are generally relevant
  - No heuristics
- Disadvantages
  - Running time cannot be guessed (finding the maximal clique is an np-complete problem)
  - Code may run for ages

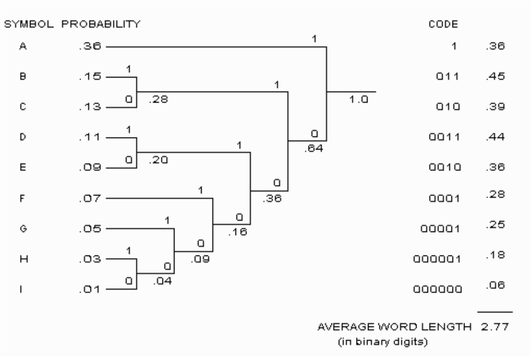
Random Walks on Graphs

- Nodes in a community have higher probability for internal than for external link.
- Random walker has a higher probability of remaining inside a community than passing to an other.
- Use this feature for community detection.
- Infomap

Infomap idea

- Take a (long) path of a random walker
- Encode it efficiently by giving unique address to each node
- Compress the encoding by assuming two level structure
- Give two level codes: Top ones (unique for each group), local (can be the same in different groups). Ex:
  - addresses in real life: Countries, Cities (there is also a Budapest in the USA), Streets (you may find Main street in many cities)
  - domain names: .hu, .de; lower domains, e.g. notebook, weather

Huffman coding



- Compress data in the most efficient general way

Huffman coding

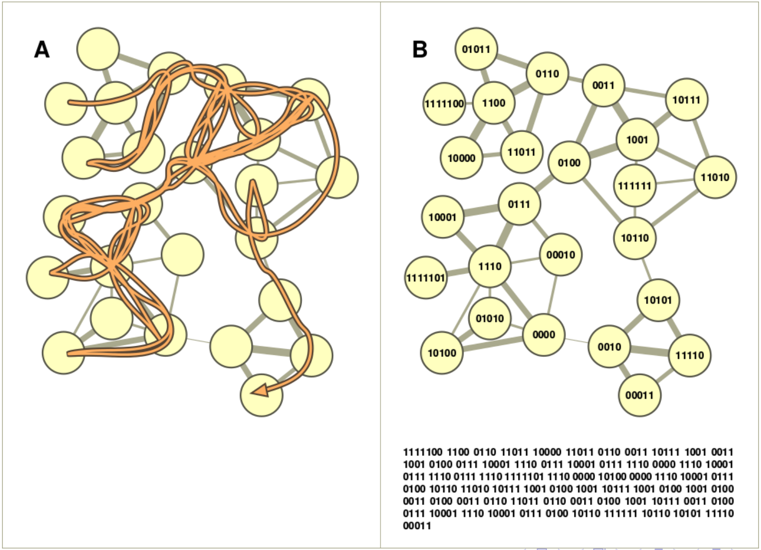
1. Create a leaf node for each symbol and add it to the priority queue.
2. While there is more than one node in the queue:
   2.1 Remove the two nodes of highest priority (lowest probability) from the queue
   2.2 Create a new internal node with these two nodes as children and with probability equal to the sum of the two nodes' probabilities.
   2.3 Add the new node to the queue.
3. The remaining node is the root node and the tree is complete.

Huffman coding, vs. infomap

- Can a coding be more efficient than Huffman coding?
- If we know more about the data yes!
- Answer: Two level coding (Of course it would be stupid for text)

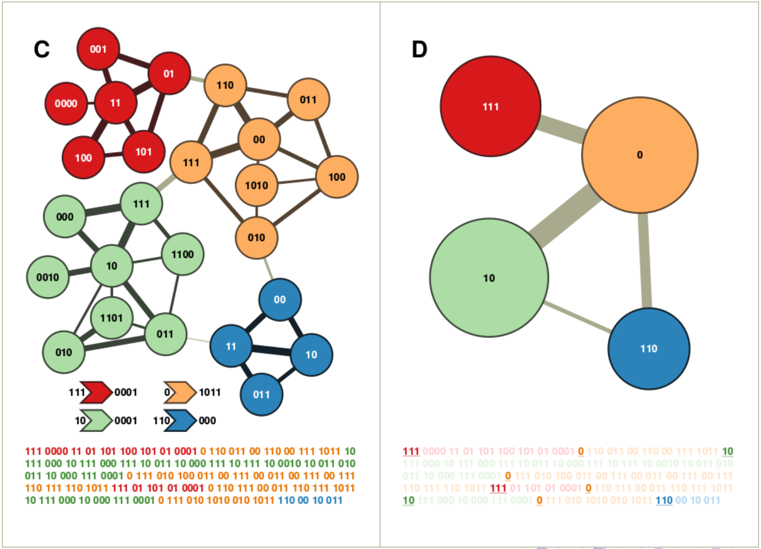
Sample random path and Huffman coding

Path length: 314 bits



Sample random path and Huffman coding

Path length: 243 bits



Infomap: Algorithm

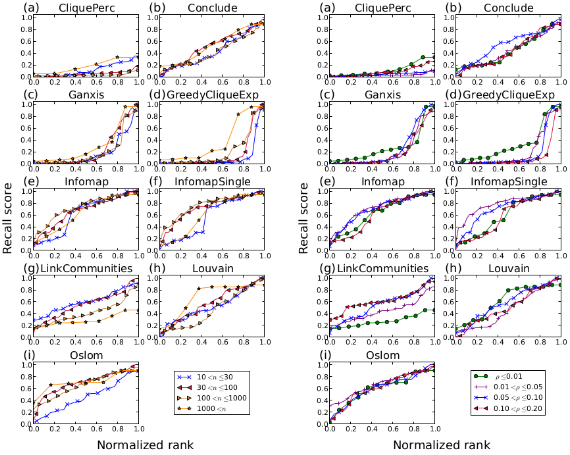
- Start with Huffman coding
- Optimize coding to minimize the *map equation*:

$$L = q_{\curvearrowright} H(Q) + \sum_{i=1}^{n_c} p_{\circlearrowright}^{i} H(P^i),$$

  where $H(Q)$ is the frequency-weighted average length of codewords for inter group jumps, $H(P^i)$ is frequency-weighted average length of codewords for group i.
- Implementation: Start with all nodes as different communities
- Merge them if L decreases

Infomap

- One of the most popular
- Fast for large networks
- Relability is comparable to more complex methods



PCA: Principal Component analysis

- Find a representative basis of the data
- Keep only relevant axes
- Represent data with coordinates on this reduced basis

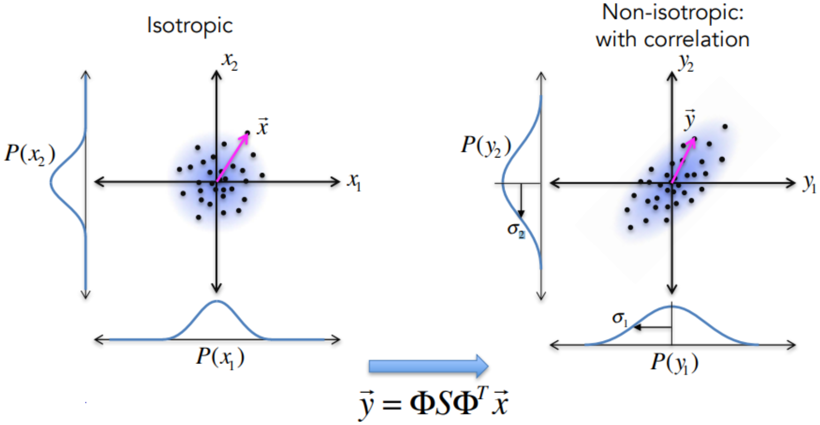


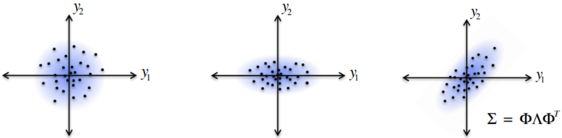
Figure from Prof. Michale Fee, MIT

PCA: Covariance matrix

- Covariance matrix:

$$\mathrm{cov}(x, y) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

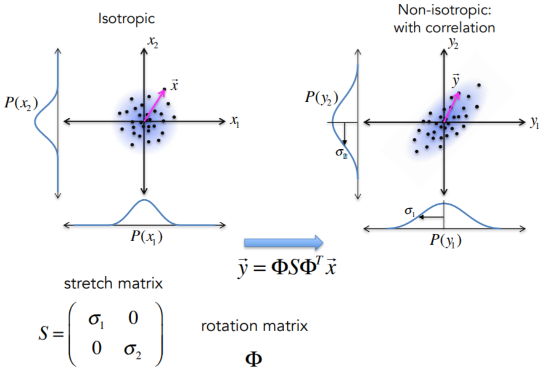
- Symmetric matrix
- If we have m variables its dimension is $m \times m$



Isotropic	Non-isotropic: no correlation	Non-isotropic: with correlation
$P(\vec{y}) = \beta e^{-\frac{1}{2\sigma^2}\vec{y}^T\vec{y}}$	$P(\vec{y}) = \beta e^{-\frac{1}{2}\vec{y}^T\Lambda^{-1}\vec{y}}$	$P(\vec{y}) = \beta e^{-\frac{1}{2}\vec{y}^T\Sigma^{-1}\vec{y}}$
σ^2	$\Lambda = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$	$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$
variance	variance matrix	covariance matrix

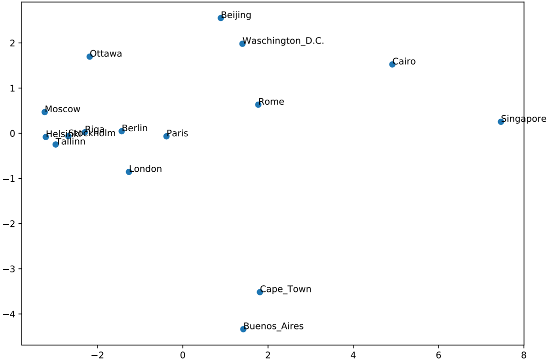
PCA: Covariance matrix

- Transformation:
- Normalized mean
- Transformation matrix
- If variances are also normalized, then transformation matrix is identity



PCA: Eigenvectors

- The covariance matrix is symmetric and positive definite
- Eigenvalues are real
- Only the first few eigenvectors are important



Practice: Hierarchical or k-means clustering

- Download the towns.dat file
- Load the file in a matrix
- Write a distance function which measures the distance in a 12 (cities) or 16 (months) dimensional space.
- Write the hierarchical, PCA or k-means clustering algorithm (do NOT use the built in ones!)
- The algorithms earn you 30, 20, 15 points respectively.