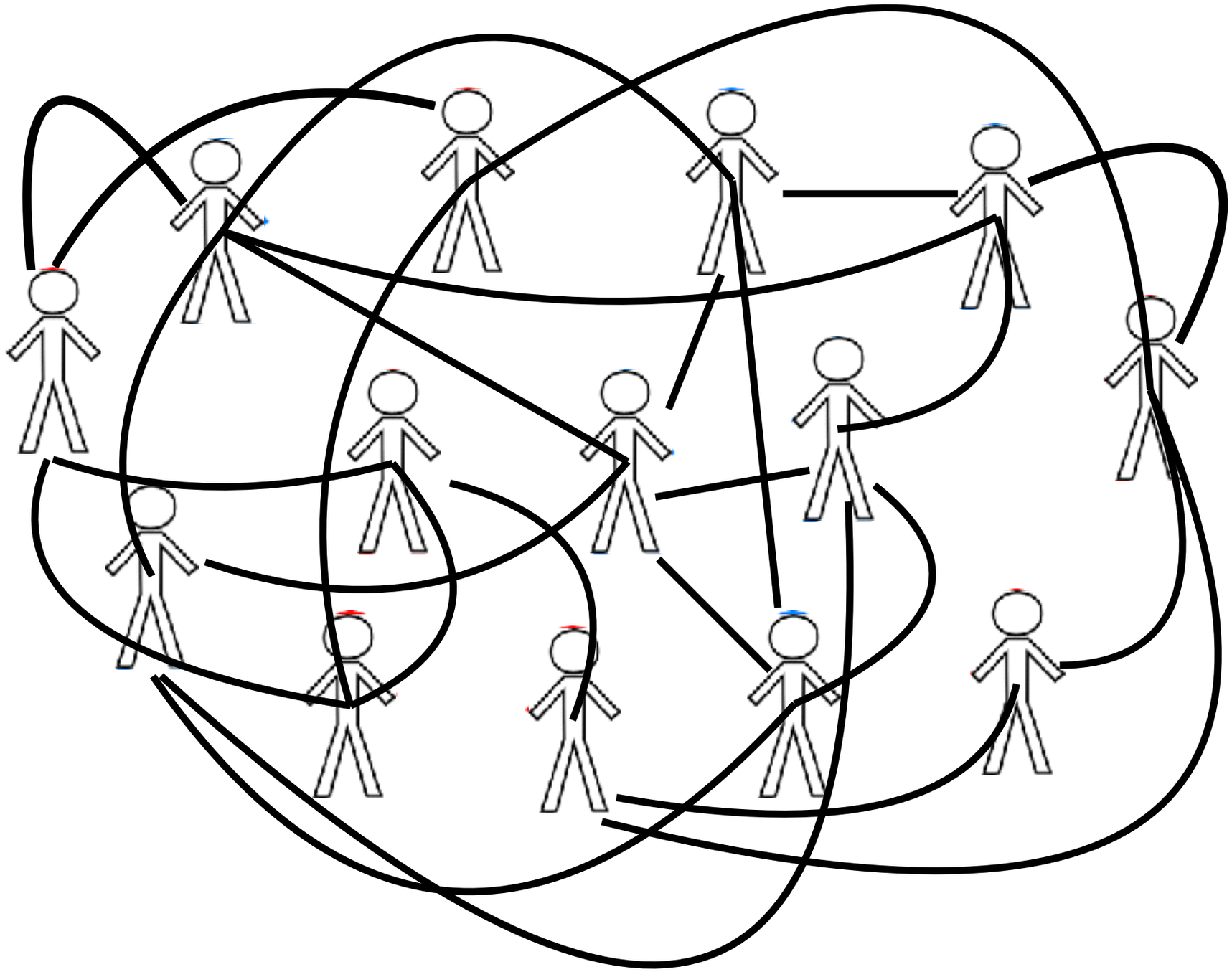
Machine Learning for Data Science (CS4786)
Lecture 11

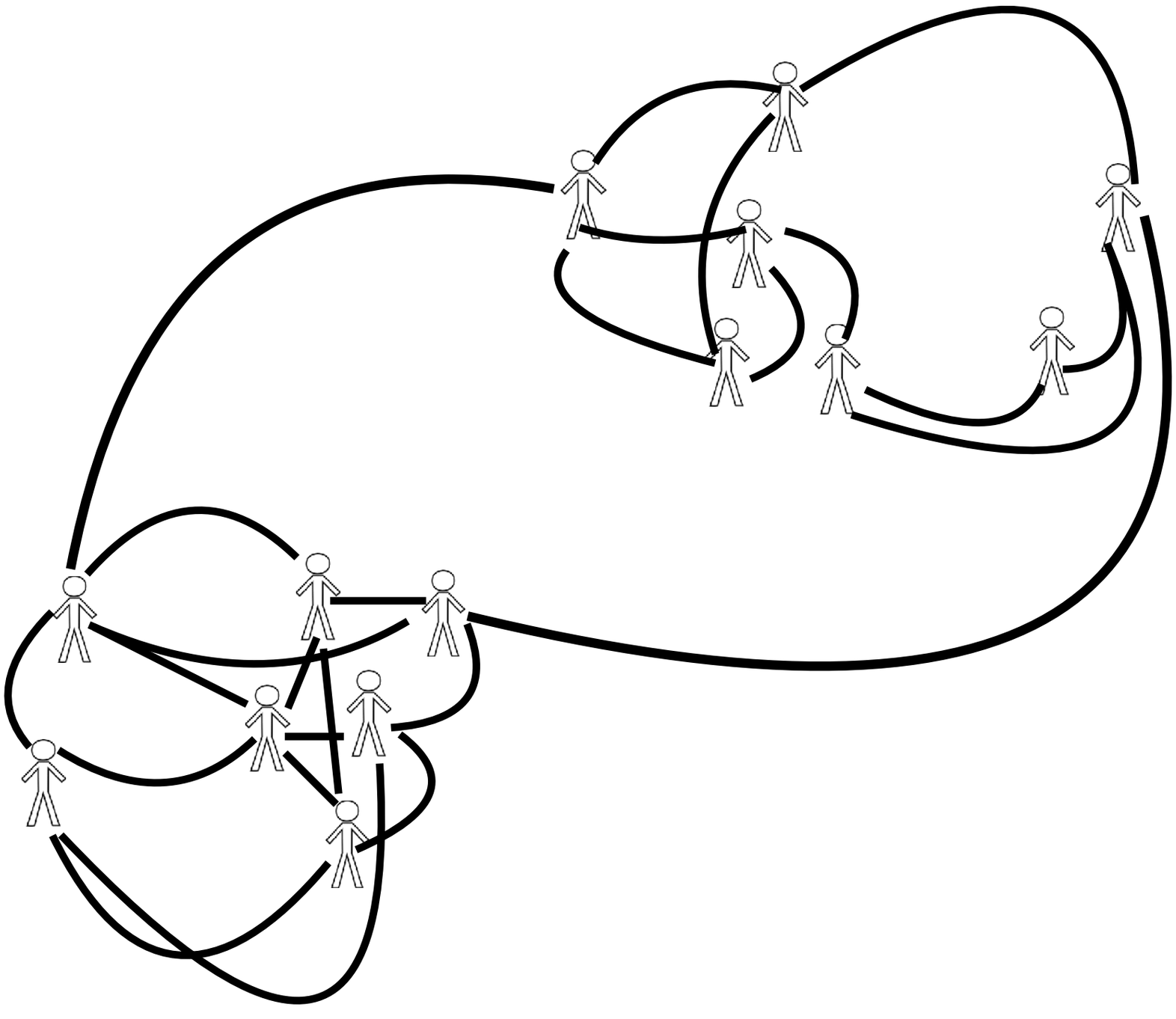
Spectral Embedding + Clustering

Motivating Example



What can you say from this network?

Motivating Example



How about now?

Thought Experiment

- For each user i we specify embedding (location) y_i
- How do we find good locations $y_1, \ldots, y_n$?
- What are good properties?

Key Principle

- **Points are centered at 0**
- **Keep your Friends close**
  (sum of distances between linked nodes should be small)
- **Variance or spread amongst the nodes should be large**

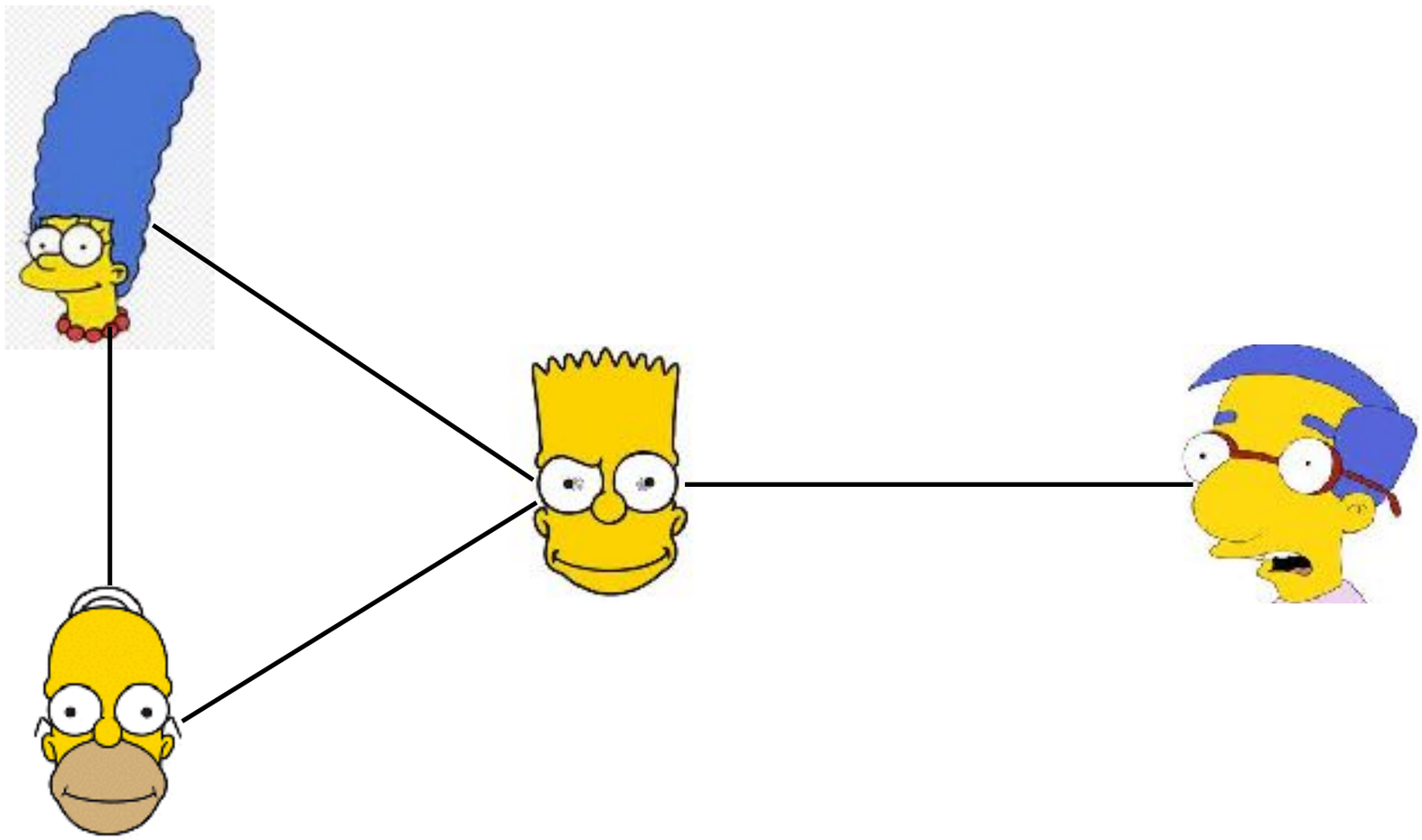
Key Principle

- **Points are centered at 0** $y^\top \mathbf{1} = 0$
- Keep your Friends close
- Variance or spread should be large

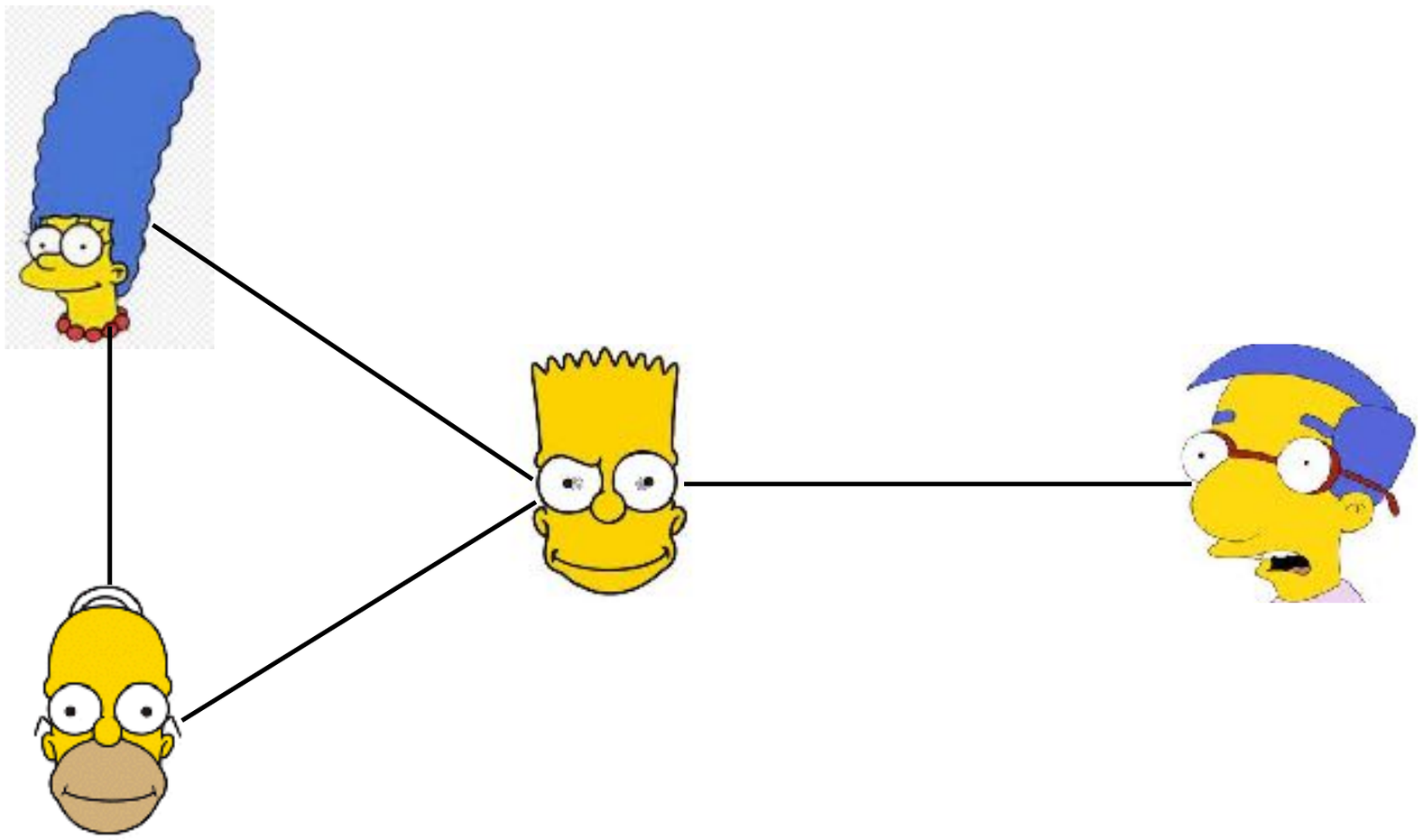
Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- **Keep your Friends close**
- Variance or spread should be large

Representing the Graph



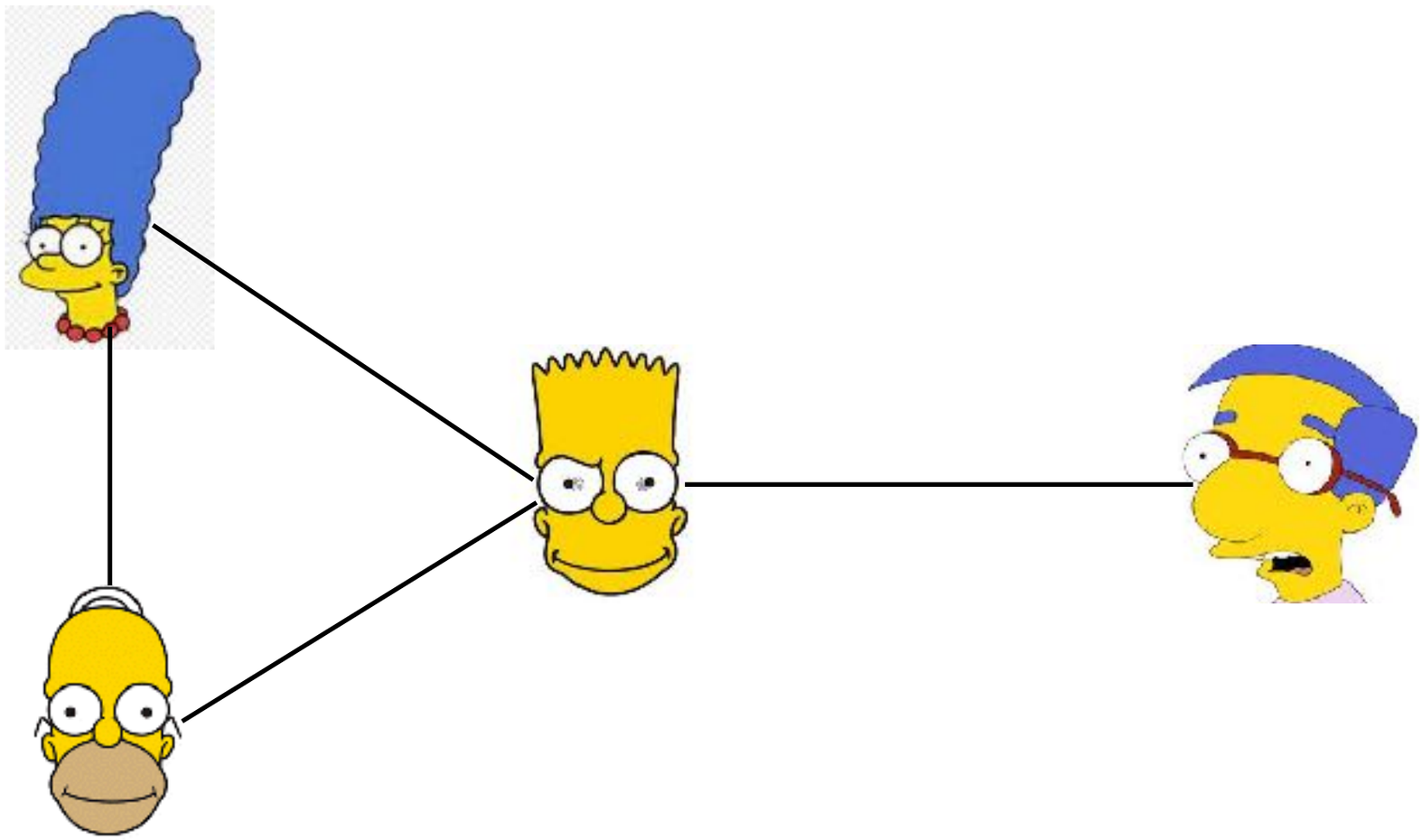
Representing the Graph



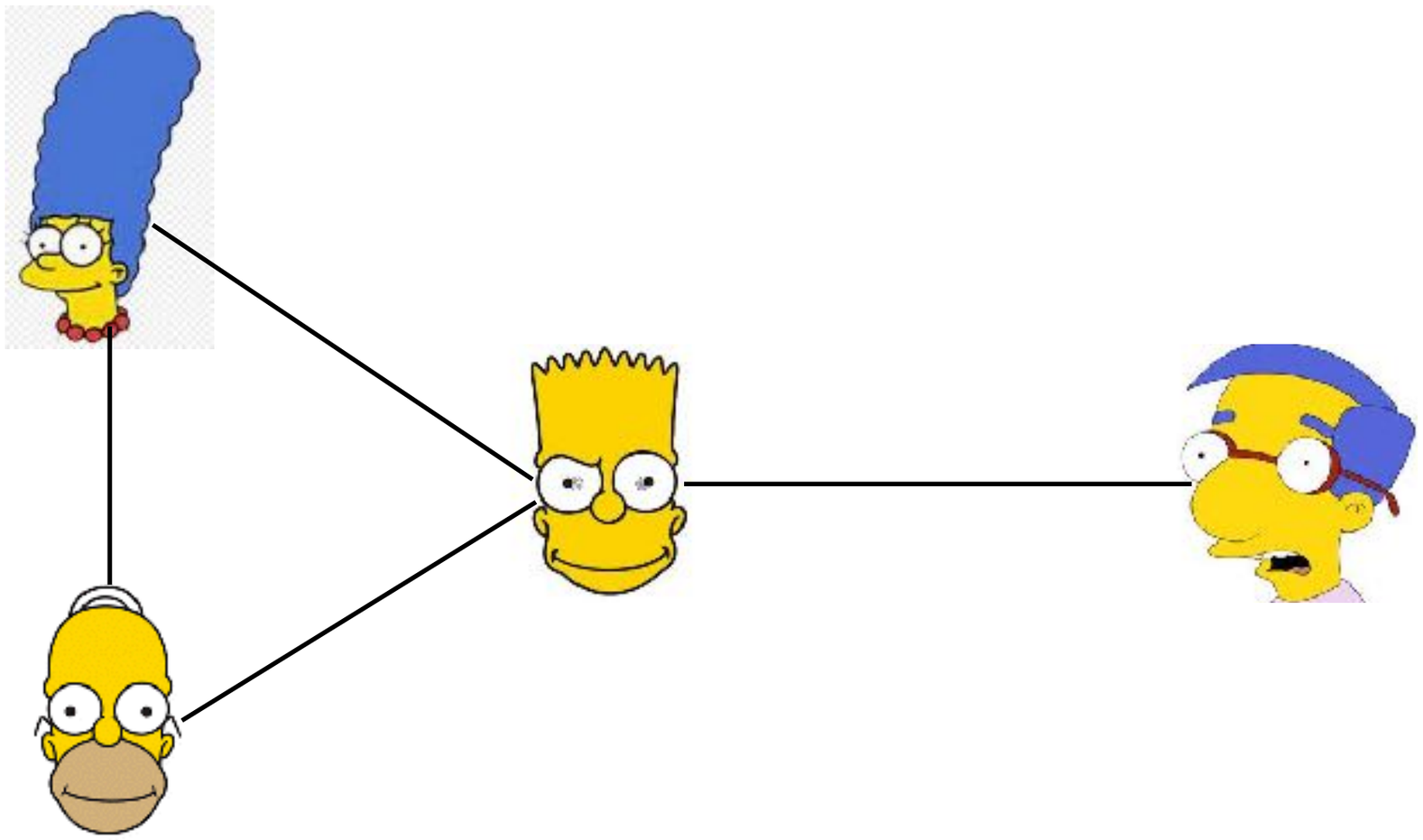
A =

	0	1	1	0
	1	0	1	0
	1	1	0	1
	0	0	1	0

Representing the Graph



Representing the Graph



D =

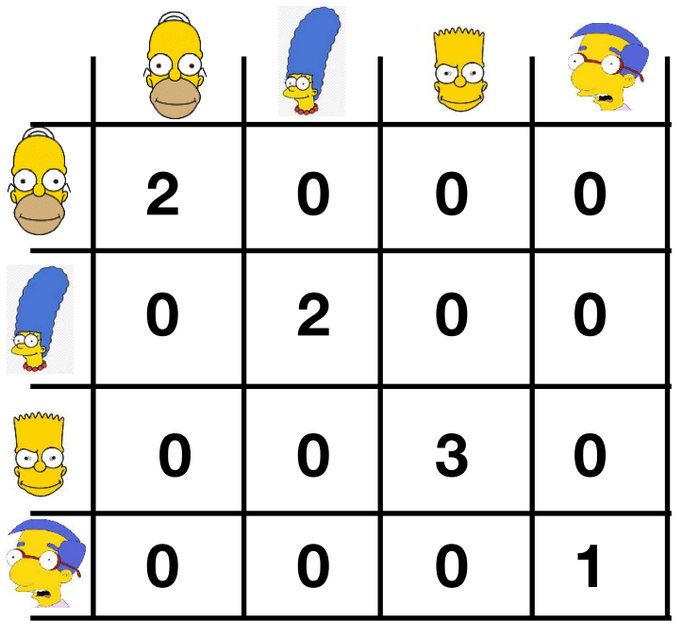
	2	0	0	0
	0	2	0	0
	0	0	3	0
	0	0	0	1

The Laplacian Matrix

$$L = D - A$$

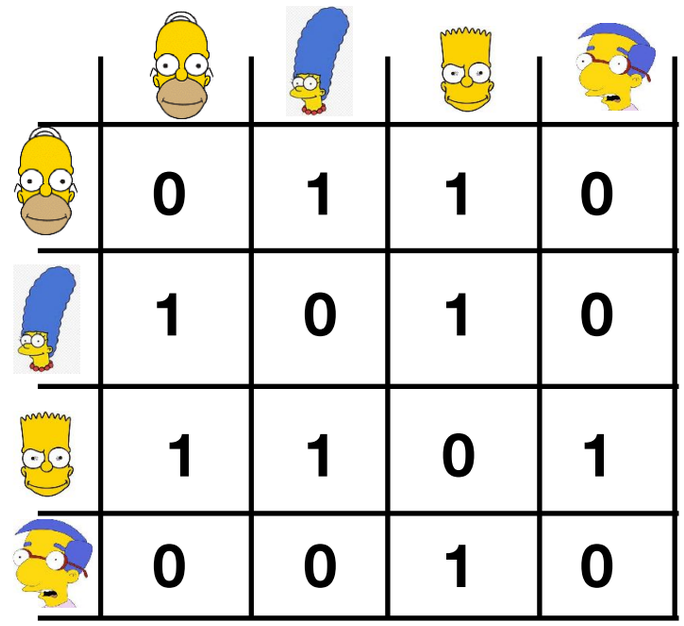
The Laplacian Matrix

$$L = D - A$$



	2	0	0	0
	0	2	0	0
	0	0	3	0
	0	0	0	1

−



	0	1	1	0
	1	0	1	0
	1	1	0	1
	0	0	1	0

Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- **Keep your Friends close**
- Variance or spread should be large

Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- **Keep your Friends close** $\text{minimize} \;\; y^\top L y$
- Variance or spread should be large

Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- **Variance or spread should be large** $\text{Maximize} \;\; \frac{1}{n}\|y\|_2^2$

Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- **Variance or spread should be large** $\text{Maximize} \;\; \frac{1}{n}\|y\|_2^2$

$$\text{Minimize} \;\; \frac{y^\top L y}{\|y\|_2^2} \quad \text{s.t. } y \perp \mathbf{1}$$

Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- **Variance or spread should be large** $\text{Maximize} \;\; \frac{1}{n}\|y\|_2^2$

$$\text{Minimize} \;\; \frac{y^\top L y}{\|y\|_2^2} \quad \text{s.t. } y \perp \mathbf{1}$$

$$\text{Minimize } y^\top L y \quad \text{s.t. } \|y\|_2^2 = 1 \quad y \perp \mathbf{1}$$

Key Principle

- Points are centered at 0 $y^\top \mathbf{1} = 0$
- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- Variance or spread should be large $\text{Maximize} \;\; \frac{1}{n}\|y\|_2^2$

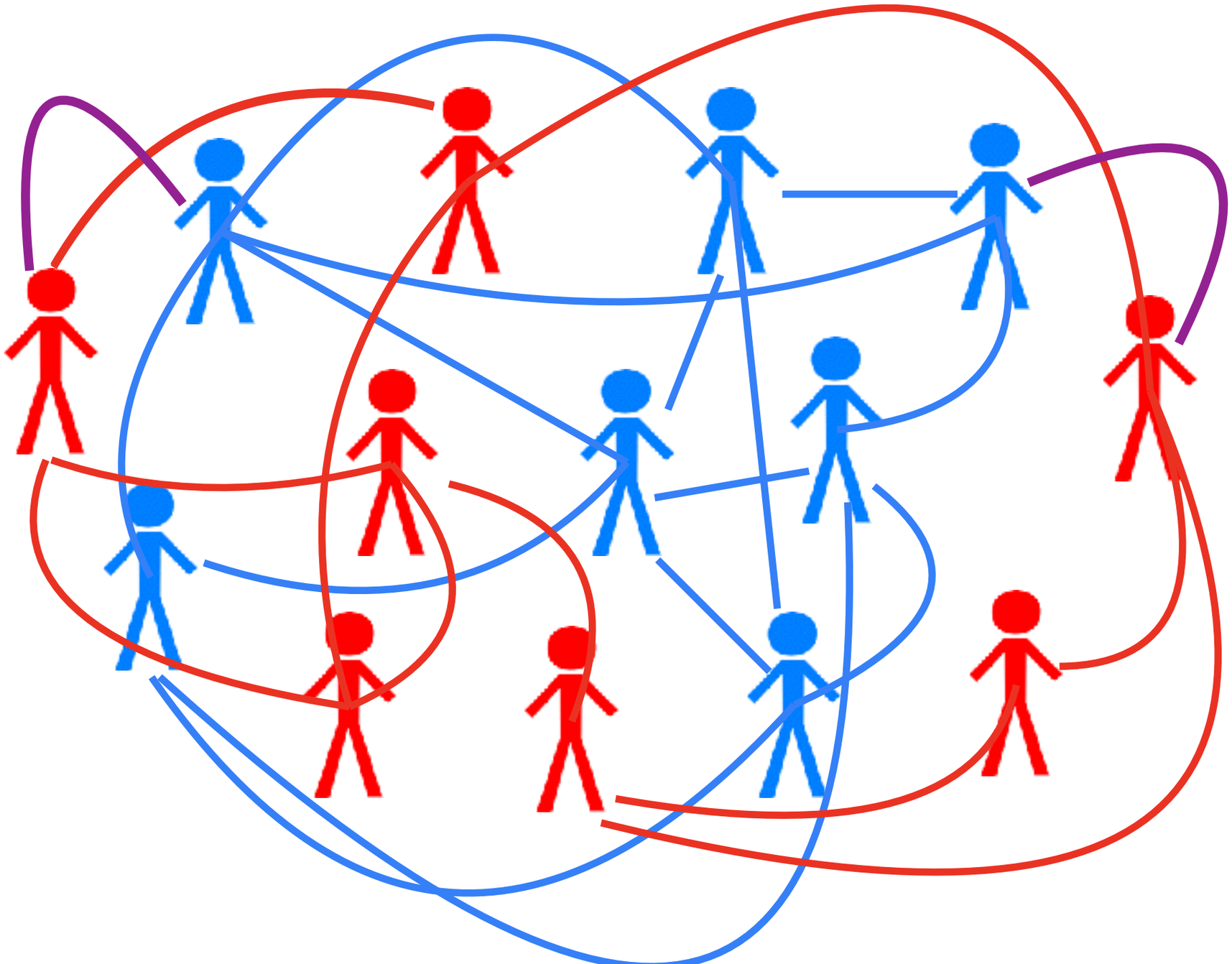
$$\text{Minimize } y^\top L y \quad \text{s.t. } \|y\|_2^2 = 1 \quad y \perp \mathbf{1}$$

$$y = \;\text{Second smallest eigenvector of } L$$

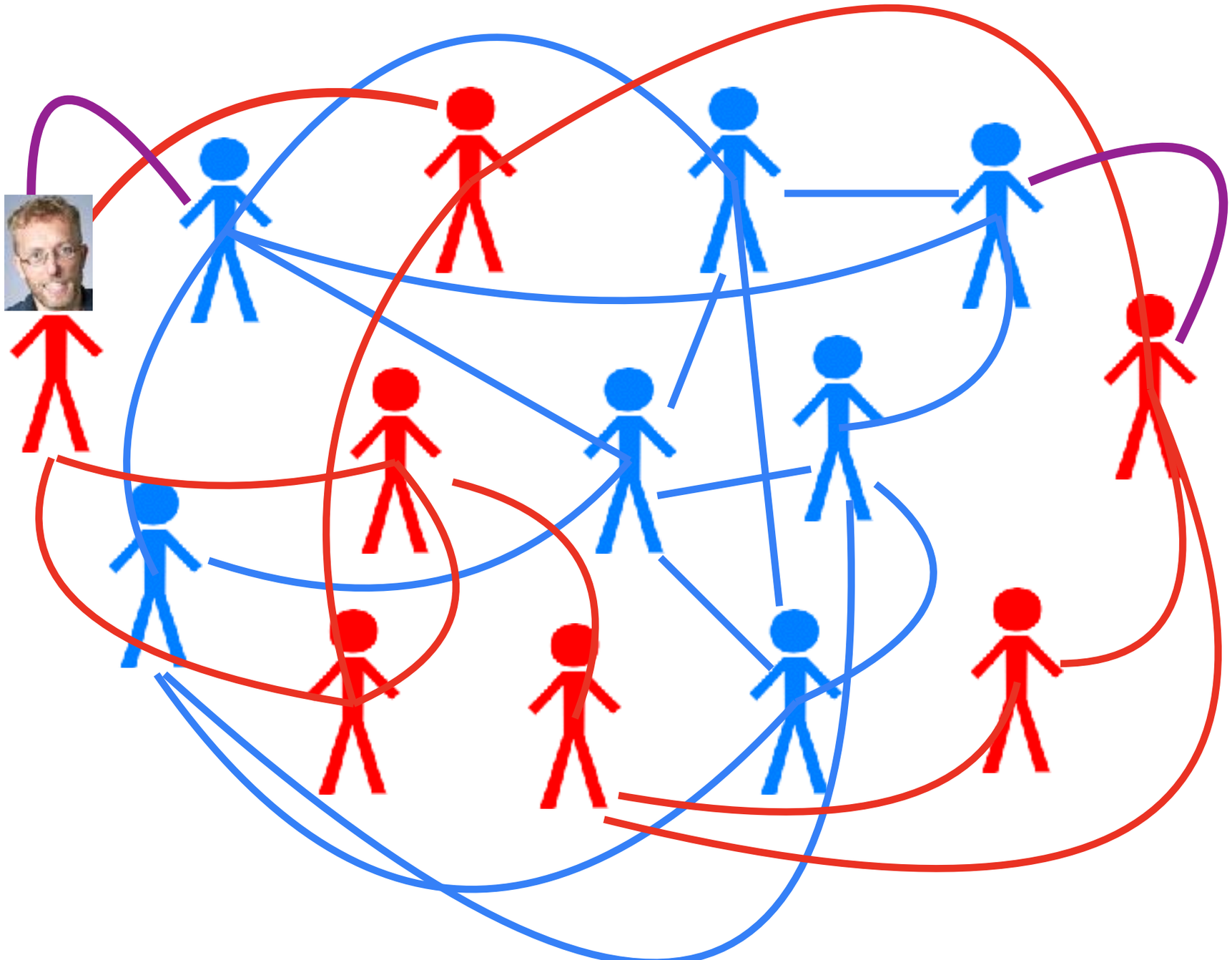
Spectral Clustering Algorithm (Unnormalized)

1. Given matrix A calculate diagonal matrix D s.t. $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$
2. Calculate the Laplacian matrix $L = D - A$
3. Find eigen vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of L (ascending order of eigenvalues)
4. Pick the K eigenvectors with smallest eigenvalues to get $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^K$
5. Use K-means clustering algorithm on $\mathbf{y}_1, \ldots, \mathbf{y}_n$

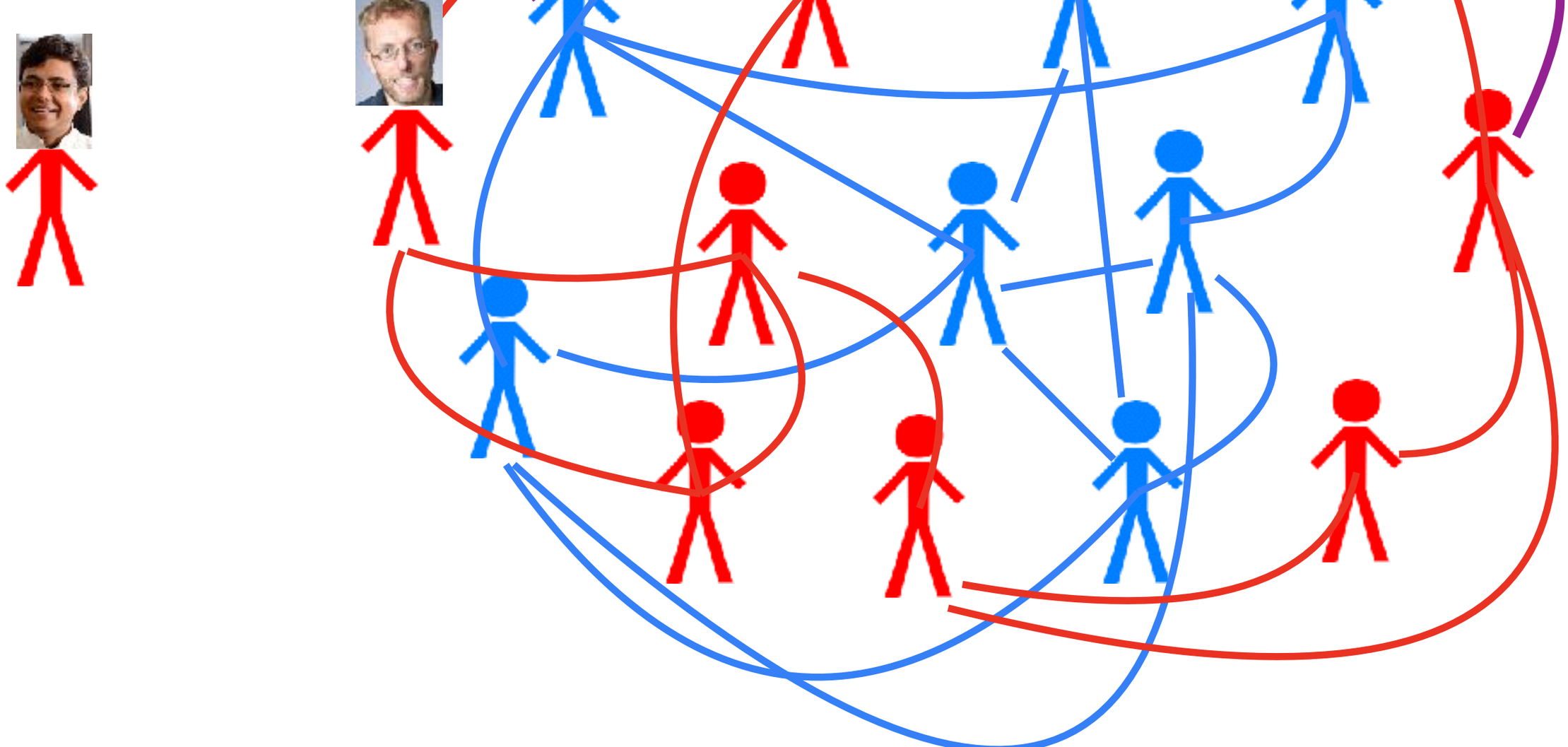
Trouble Makers



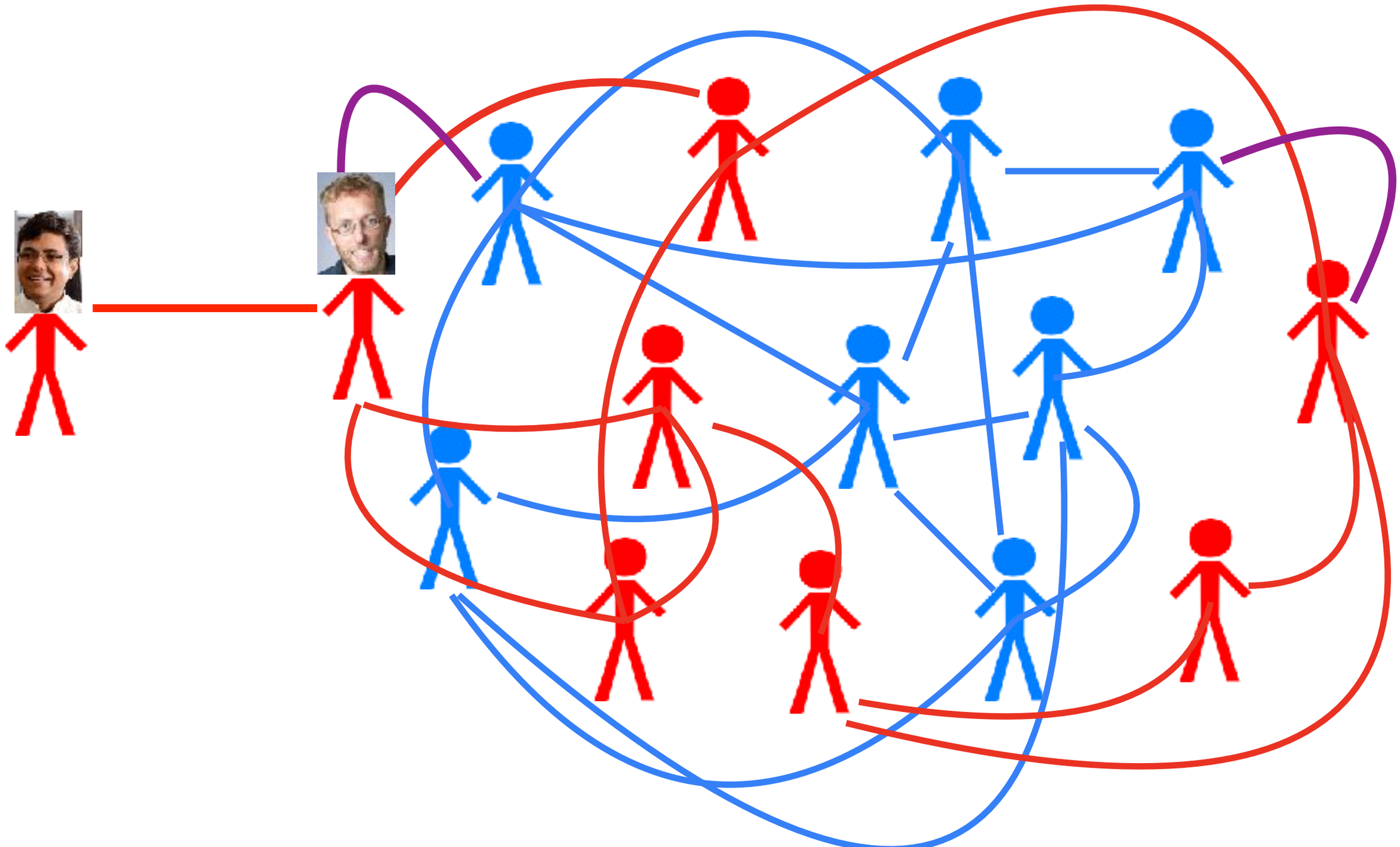
Trouble Makers



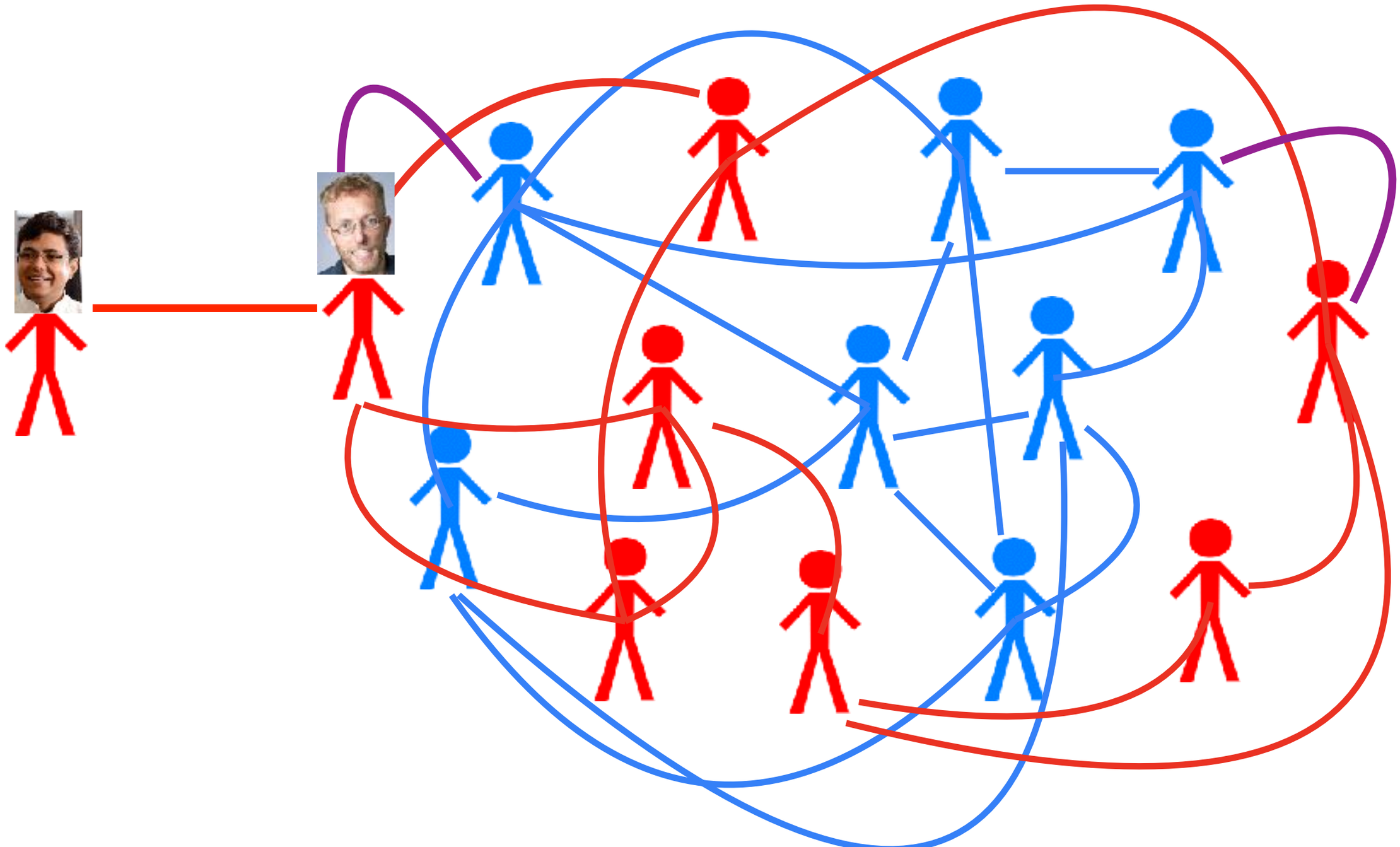
Trouble Makers



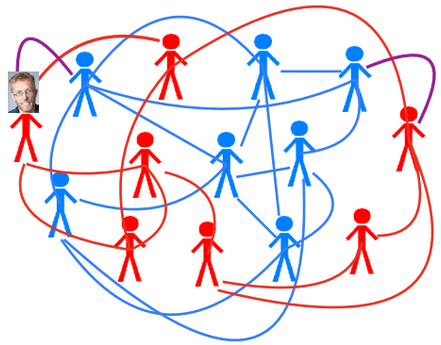
Trouble Makers



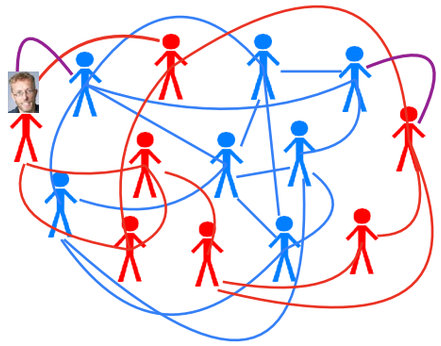
Trouble Makers



Trouble Makers

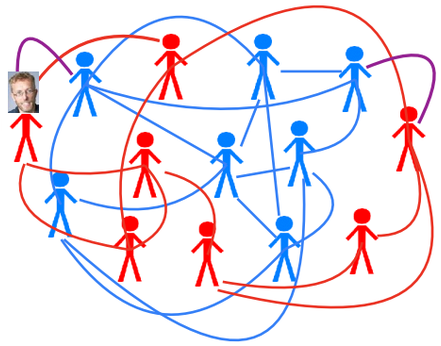


Trouble Makers



- Variance is high

Trouble Makers



- Variance is high
- Almost all connected nodes have same (small value)

Demo

What was the problem?

What was the problem?

- Pushing few nodes far away from the rest increased variance

What was the problem?

- Pushing few nodes far away from the rest increased variance
- But these nodes were ones with one or few links

What was the problem?

- Pushing few nodes far away from the rest increased variance

- But these nodes were ones with one or few links

- We want nodes with fewer links to account for lesser of the variance

Normalized Spectral Embedding

Normalized Spectral Embedding

- Nodes linked to each other are close to each other

Normalized Spectral Embedding

- Nodes linked to each other are close to each other
- Variance or spread should be large

Normalized Spectral Embedding

- Nodes linked to each other are close to each other
- Variance or spread should be large
  - But variance under what distribution?

Normalized Spectral Embedding

- Nodes linked to each other are close to each other
- Variance or spread should be large
  - But variance under what distribution?
    - Higher degree nodes are more important!

Normalized Spectral Embedding

- Nodes linked to each other are close to each other
- Variance or spread should be large
  - But variance under what distribution?
    - Higher degree nodes are more important!
    - Lets try a distribution where probability of picking node is proportional to its degree

Normalized Spectral Embedding

- Nodes linked to each other are close to each other
- Variance or spread should be large
  - But variance under what distribution?
    - Higher degree nodes are more important!
    - Lets try a distribution where probability of picking node is proportional to its degree
      - so pushing loners away does not account for much variance

Normalized Spectral Embedding

- Nodes linked to each other are close to each other
- Variance or spread should be large
  - But variance under what distribution?
    - Higher degree nodes are more important!
    - Lets try a distribution where probability of picking node is proportional to its degree
      - so pushing loners away does not account for much variance
      - There is more incentive to push the famous people outwards

Normalized Spectral Embedding

Normalized Spectral Embedding

Define distribution with $p_i = \dfrac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \dfrac{D_{i,i}}{|E|}$

Normalized Spectral Embedding

$$\text{Define distribution with } p_i = \frac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \frac{D_{i,i}}{|E|}$$

- Keep your Friends close

Normalized Spectral Embedding

$$\text{Define distribution with } p_i = \frac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \frac{D_{i,i}}{|E|}$$

- Keep your Friends close $\quad \text{minimize } \; y^\top L y$

Normalized Spectral Embedding

$$\text{Define distribution with } p_i = \frac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \frac{D_{i,i}}{|E|}$$

- Keep your Friends close $\quad \text{minimize} \;\; y^\top L y$
- Points are centered at 0

Normalized Spectral Embedding

$$\text{Define distribution with } p_i = \frac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \frac{D_{i,i}}{|E|}$$

- Keep your Friends close $\underset{n}{\text{minimize}} \; y^\top L y$
- Points are centered at 0 $\sum_{i=1}^{n} p_i y_i = 0 \qquad \sum_{i=1}^{n} D_{i,i} y_i = 0$

Normalized Spectral Embedding

$$\text{Define distribution with } p_i = \frac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \frac{D_{i,i}}{|E|}$$

- Keep your Friends close $\underset{n}{\text{minimize}} \; y^\top L y$
- Points are centered at 0 $\sum_{i=1}^{n} p_i y_i = 0 \qquad \sum_{i=1}^{n} D_{i,i} y_i = 0$
- Variance or spread should be large

Normalized Spectral Embedding

$$\text{Define distribution with } p_i = \frac{D_{i,i}}{\sum_{j=1}^{n} D_{j,j}} = \frac{D_{i,i}}{|E|}$$

- Keep your Friends close $\underset{y \in \mathbb{R}^n}{\text{minimize}} \; y^\top L y$
- Points are centered at 0 $\sum_{i=1} p_i y_i = 0 \qquad \sum_{i=1} D_{i,i} y_i = 0$
- Variance or spread should be large

$$\text{Maximize} \sum_{i=1}^{n} p_i y_i^2 = \frac{1}{|E|} \sum_{i=1}^{n} D_{i,i} y_i^2 = \frac{1}{|E|} y^\top D y$$

Normalized Spectral Embedding

- Keep your Friends close $\quad \text{minimize} \quad y^\top L y$
- Points are centered at 0 $\quad \sum_{i=1}^{n} D_{i,i} y_i = 0$
- Variance or spread should be large $\quad \text{Maximize} \quad y^\top D y$

Normalized Spectral Embedding

- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- Points are centered at 0 $\sum_{i=1}^{n} D_{i,i} y_i = 0$
- Variance or spread should be large $\text{Maximize} \;\; y^\top D y$

$$\text{Minimize} \;\; \frac{y^\top L y}{y^\top D y} \quad \text{s.t.} \;\; \sum_{i=1}^{n} D_{i,i} y_i = 0$$

Normalized Spectral Embedding

- Keep your Friends close $\quad \text{minimize} \quad y^\top L y$
- Points are centered at 0 $\quad \sum_{i=1}^{n} D_{i,i} y_i = 0$
- Variance or spread should be large $\quad \text{Maximize} \quad y^\top D y$

$$\text{Minimize} \quad \frac{y^\top L y}{y^\top D y} \quad \text{s.t.} \quad \sum_{i=1}^{n} D_{i,i} y_i = 0$$

$$\text{Define } u = D^{1/2} y \text{ so that } \frac{y^\top L y}{y^\top D y} = \frac{u^\top D^{-1/2} L D^{-1/2} u}{\|u\|^2}$$

Normalized Spectral Embedding

- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- Points are centered at 0 $\sum_{i=1}^{n} D_{i,i} y_i = 0$
- Variance or spread should be large $\text{Maximize} \;\; y^\top D y$

$$\text{Minimize} \;\; \frac{y^\top L y}{y^\top D y} \quad \text{s.t.} \;\; \sum_{i=1}^{n} D_{i,i} y_i = 0$$

$$\text{Define } u = D^{1/2} y \text{ so that } \frac{y^\top L y}{y^\top D y} = \frac{u^\top D^{-1/2} L D^{-1/2} u}{\|u\|^2}$$

$$\text{and } \sum_{i=1}^{n} D_{i,i} y_i = \sum_{i=1}^{n} D_{i,i}^{1/2} u_i = 0 \Rightarrow \text{diag}(D^{1/2}) \perp u$$

Normalized Spectral Embedding

- Keep your Friends close $\text{minimize} \;\; y^\top L y$
- Points are centered at 0 $\sum_{i=1}^{n} D_{i,i} y_i = 0$
- Variance or spread should be large $\text{Maximize} \;\; y^\top D y$

$$\text{Minimize } u^\top D^{-1/2} L D^{-1/2} u \qquad \text{s.t. } \|u\| = 1 \;\&\; u \perp \text{diag}(D^{1/2})$$

Solution: Second smallest eigen vector of $D^{-1/2} L D^{-1/2}$

Normalized Spectral Embedding

- More generally if probability of a node i is proportional to some p_i then solution to normalized spectral clustering is
  - Second smallest eigen vector of $P^{-1/2}LP^{-1/2}$ where $P = \mathrm{diag}(p)$
    - For K dimensional representation, we can take 2nd to K+1'th smallest eigenvectors say $u_{2,\ldots,} u_{K+1}$
- $Y = P^{-1/2}U$

Demo

Clustering

Clustering

- Grouping sets of data points s.t.
  - points in same group are similar
  - points in different groups are dissimilar

- A form of unsupervised classification where there are no predefined labels

Clustering

- Partition data into K disjoint groups

Clustering

- Partition data into K disjoint groups
- Compression or Quantization
  - Compress n points into K representatives/groups

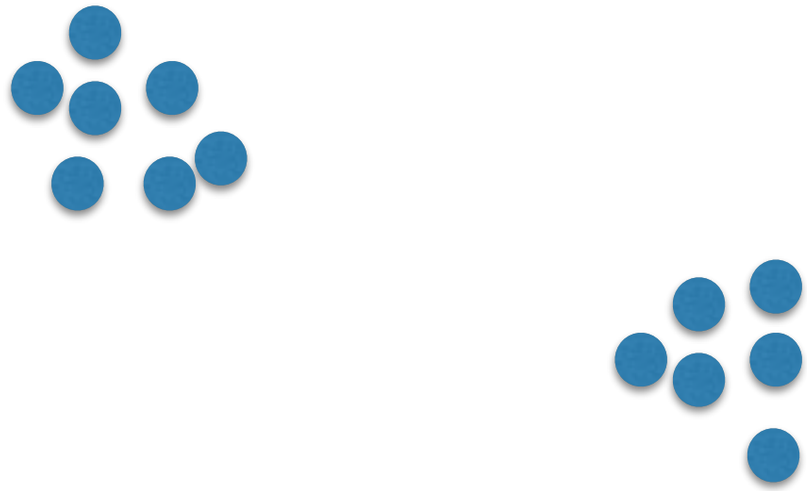
Clustering

- Partition data into K disjoint groups
- Compression or Quantization
  - Compress n points into K representatives/groups
- Visualization or Understanding
  - Taxonomy: Animals Vs plants Vs Microbes, Science Vs Math Vs Social Sciences
  - Segmentation: different types of customers, students etc. Find natural groupings in data

Clustering

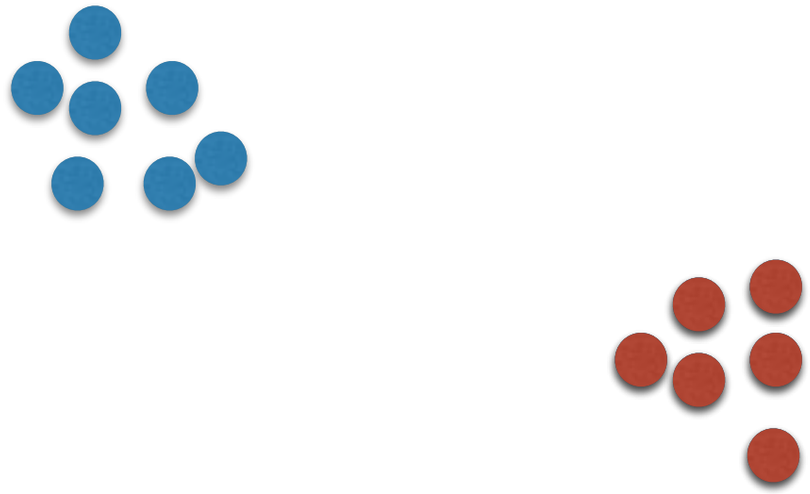
- Partition data into K disjoint groups
- Compression or Quantization
  - Compress n points into K representatives/groups
- Visualization or Understanding
  - Taxonomy: Animals Vs plants Vs Microbes, Science Vs Math Vs Social Sciences
  - Segmentation: different types of customers, students etc. Find natural groupings in data
- What this does not include: items belonging to more than one type

Examples



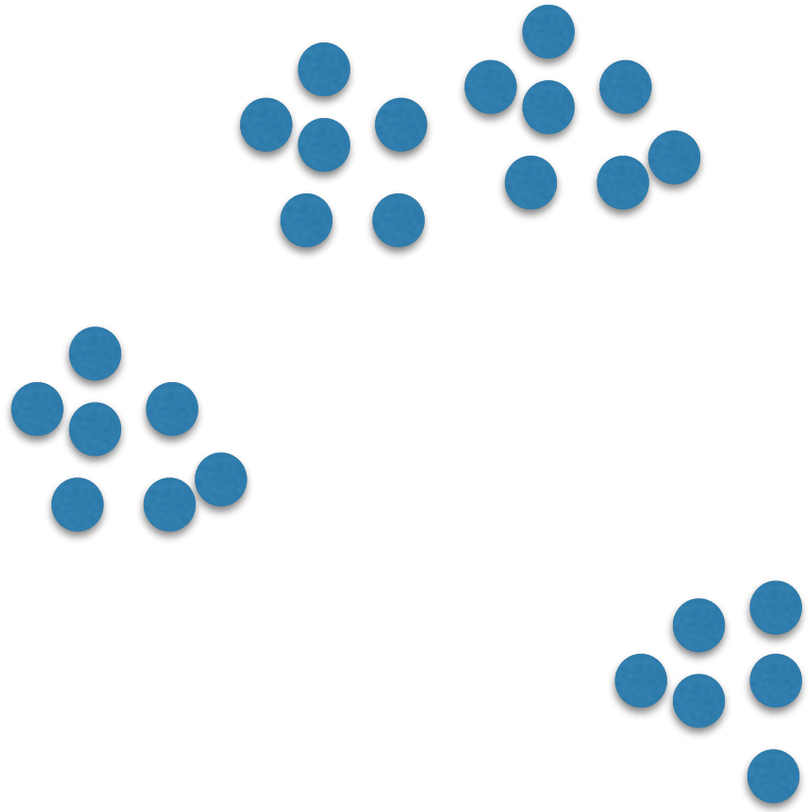
What are the clusters?

Examples



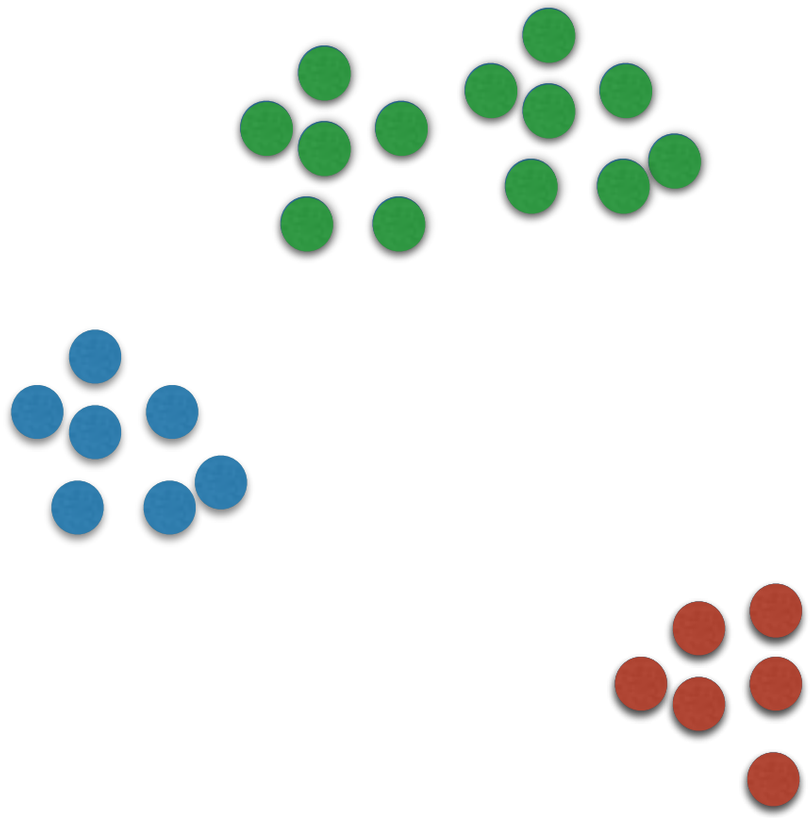
What are the clusters?

Examples



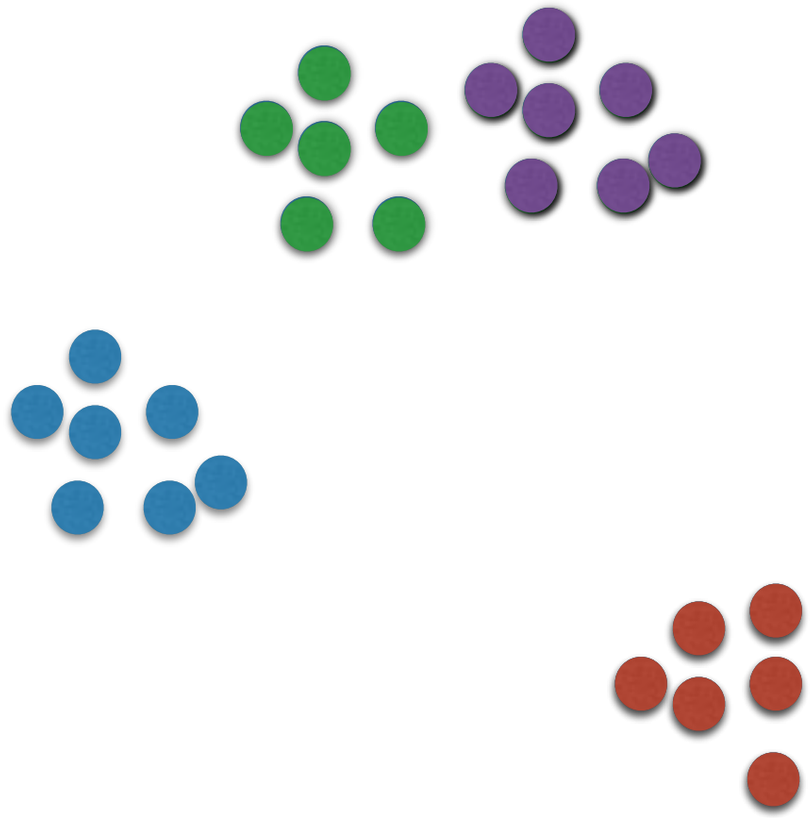
What are the clusters?

Examples



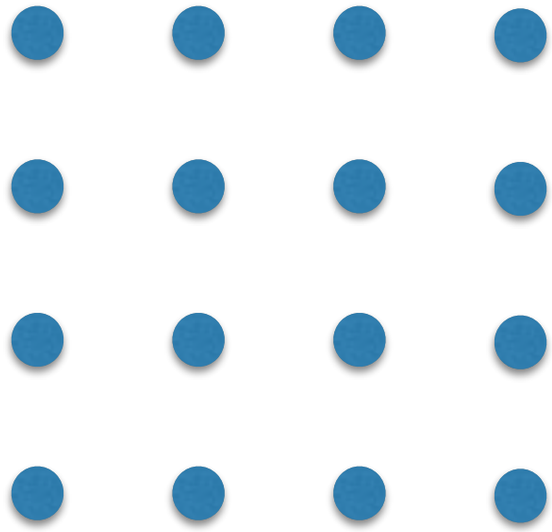
What are the clusters?

Examples



What are the clusters?

Examples



What are the clusters?

Clustering

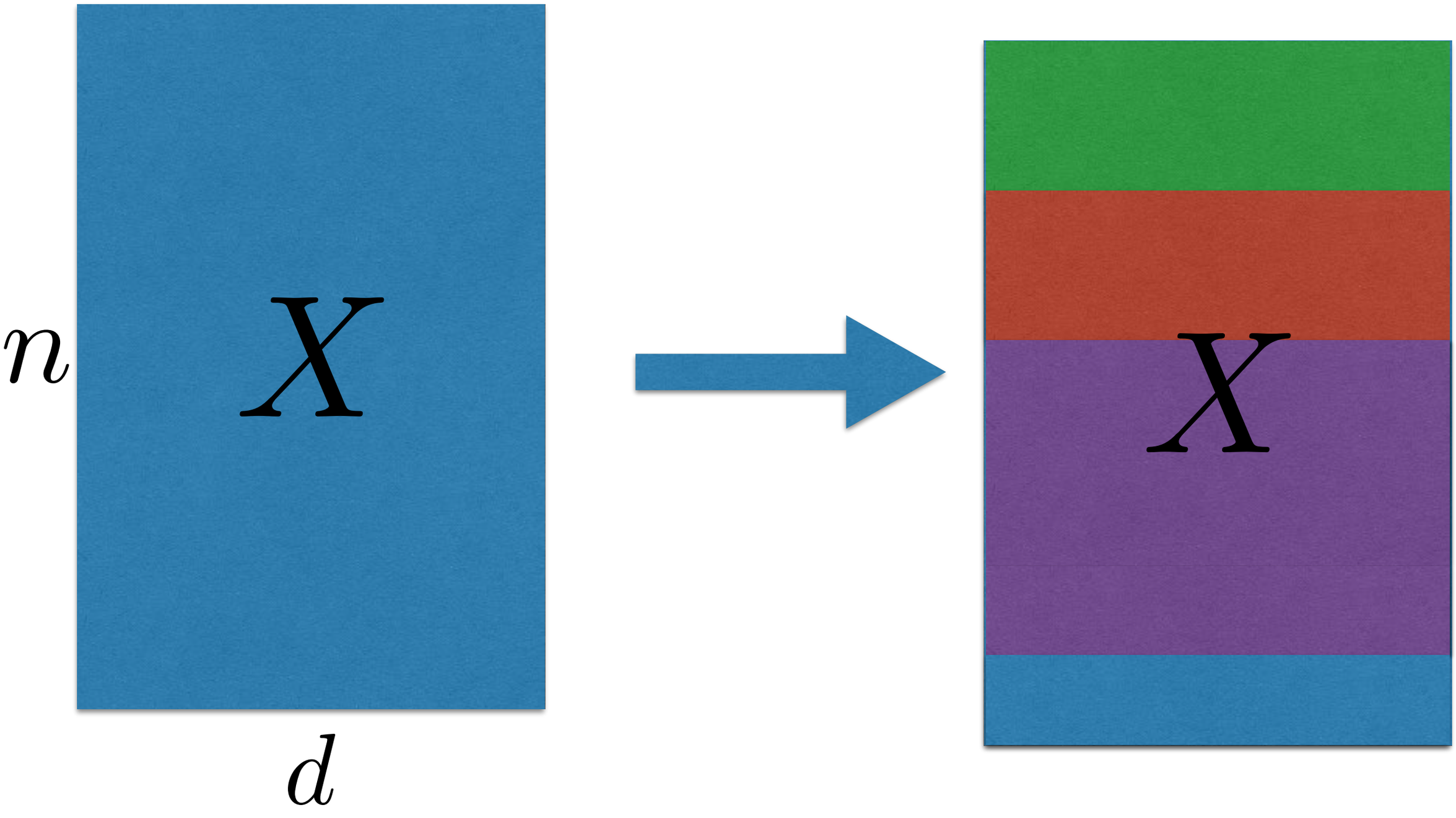
Clustering

n

X

d

Clustering



Clustering

- Grouping sets of data points s.t.
  - points in same group are similar
  - points in different groups are dissimilar

- A form of unsupervised classification where there are no predefined labels

Some Notations

- Kary clustering is a partition of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into K groups
- For now assume the magical K is given to use
- Clustering given by $C_1, \ldots, C_K$, the partition of data points.
- Given a clustering, we shall use $c(\mathbf{x}_t)$ to denote the cluster identity of point $\mathbf{x}_t$ according to the clustering.
- Let n_j denote $|C_j|$, clearly $\sum_{j=1}^{K} n_j = n$.

How do we formalize a good clustering objective?

How do we formalize?

Say dissimilarity$(\mathbf{x}_t, \mathbf{x}_s)$ measures dissimilarity between $\mathbf{x}_t$ & $\mathbf{x}_s$

Given two clustering $\{C_1, \ldots, C_K\}$ (or c) and $\{C'_1, \ldots, C'_K\}$ (or c')

How do we decide which is better?

How do we formalize?

Say dissimilarity$(\mathbf{x}_t, \mathbf{x}_s)$ measures dissimilarity between $\mathbf{x}_t$ & $\mathbf{x}_s$

Given two clustering $\{C_1, \ldots, C_K\}$ (or c) and $\{C'_1, \ldots, C'_K\}$ (or c')

How do we decide which is better?

- points in same cluster are not dissimilar
- points in different clusters are dissimilar

Clustering Criterion

- Minimize total within-cluster dissimilarity

Clustering Criterion

- Minimize total within-cluster dissimilarity

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

Clustering Criterion

- Minimize total within-cluster dissimilarity

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

- Maximize between-cluster dissimilarity

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t : C(\mathbf{x}_s) \neq C(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

Clustering Criterion

- Minimize total within-cluster dissimilarity

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

- Maximize between-cluster dissimilarity

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Maximize smallest between-cluster dissimilarity

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

Clustering Criterion

- Minimize total within-cluster dissimilarity

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

- Maximize between-cluster dissimilarity

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Maximize smallest between-cluster dissimilarity

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Minimize largest within-cluster dissimilarity

$$M_4 = \max_{j \in [K]} \max_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

Clustering Criterion

- Minimize total within-cluster dissimilarity

$$M_1 = \sum_{j=1}^{K} \sum_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$

- Maximize between-cluster dissimilarity

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Maximize smallest between-cluster dissimilarity

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(x_t, x_s)$$

- Minimize largest within-cluster dissimilarity

$$M_4 = \max_{j \in [K]} \max_{s,t \in C_j} \text{dissimilarity}(x_t, x_s)$$