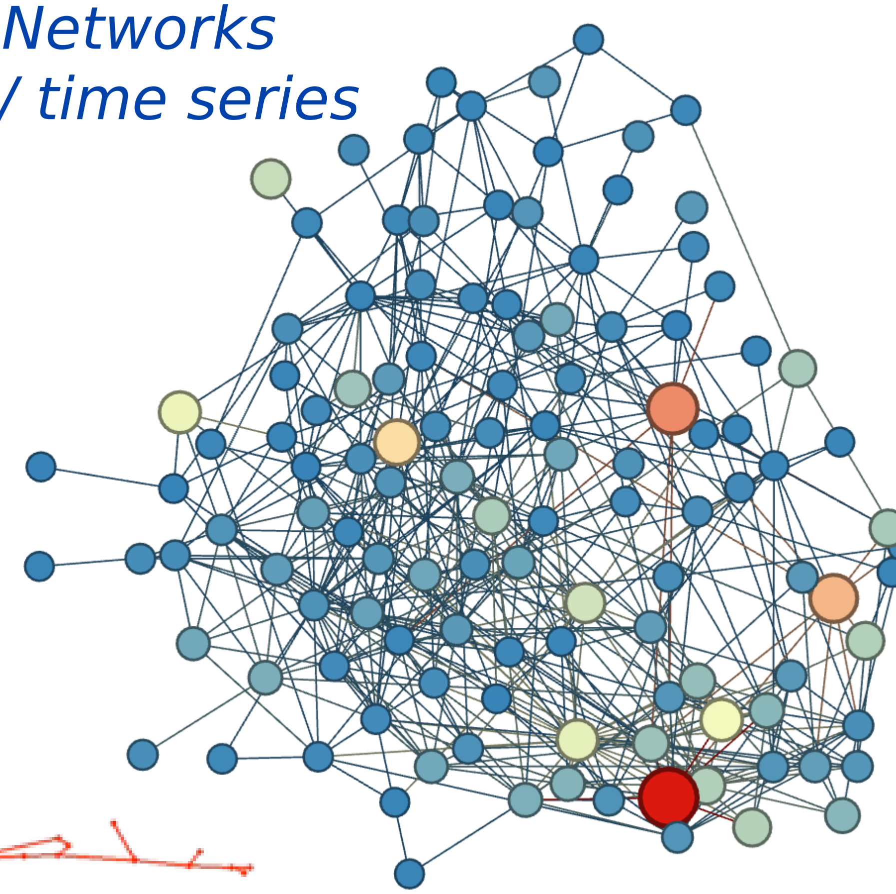
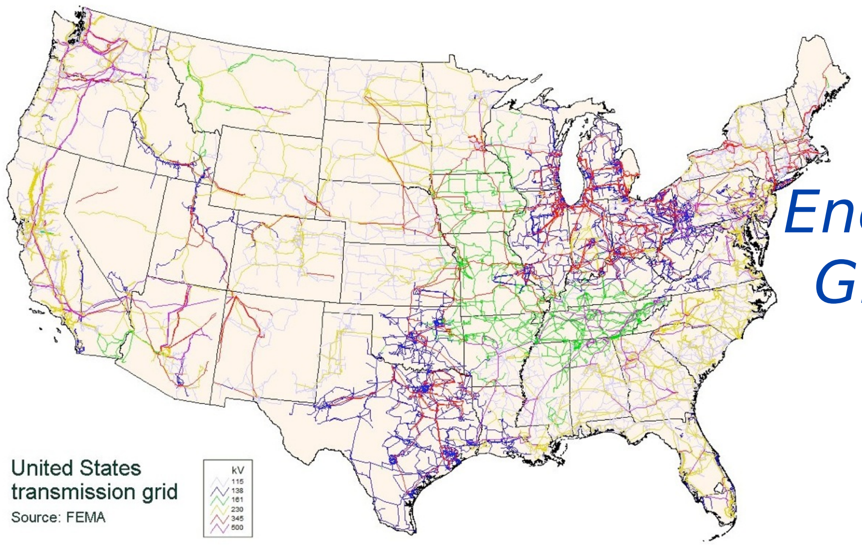
Learning and signal processing on graphs

Benjamin Ricaud, LTS2, EPFL

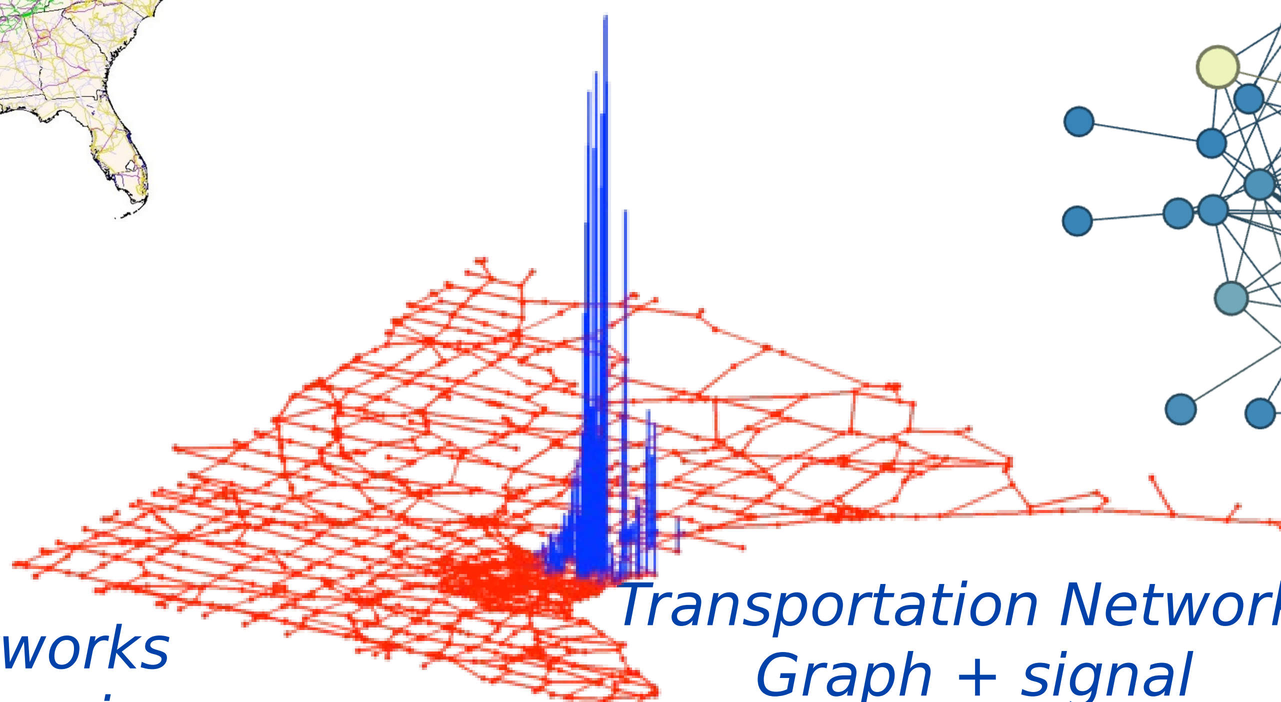
Processing Data on/with Graphs

Web or Social Networks
Graph + activity / time series



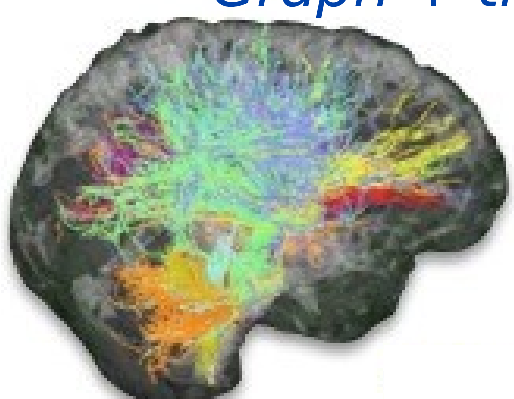


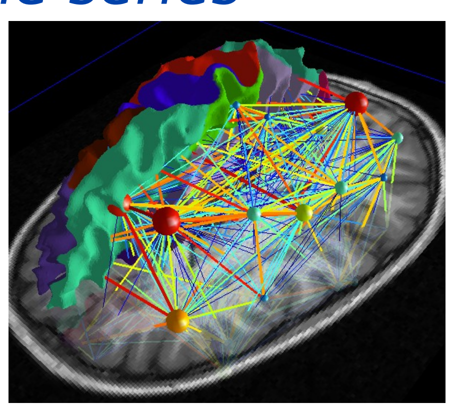
Energy Networks
Graph + signal



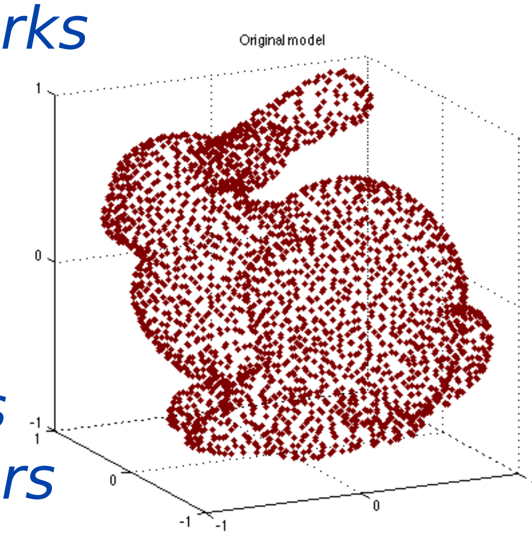
Transportation Networks
Graph + signal

Biological Networks
Graph + time series



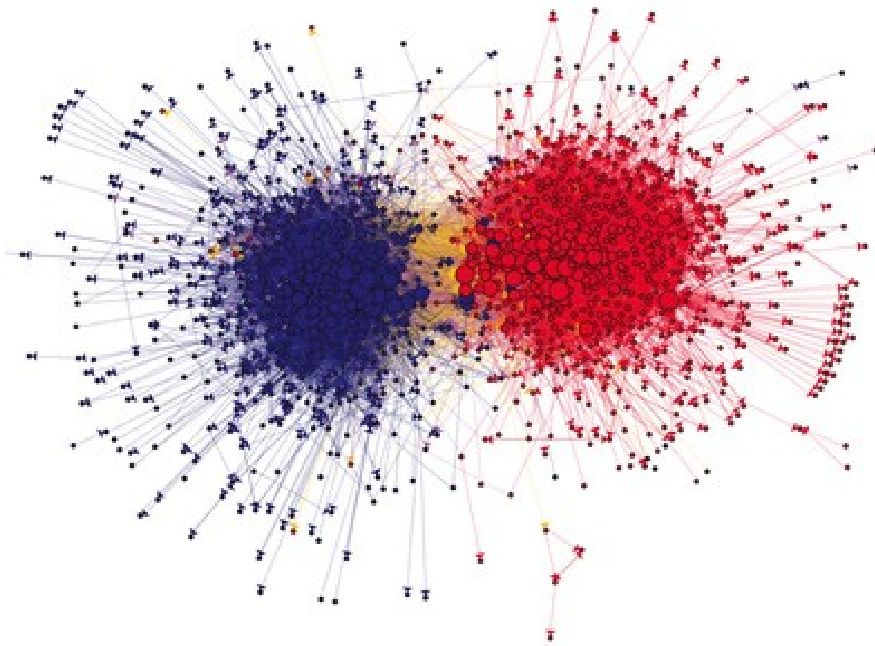


Point clouds + colors
Graph + feature vectors

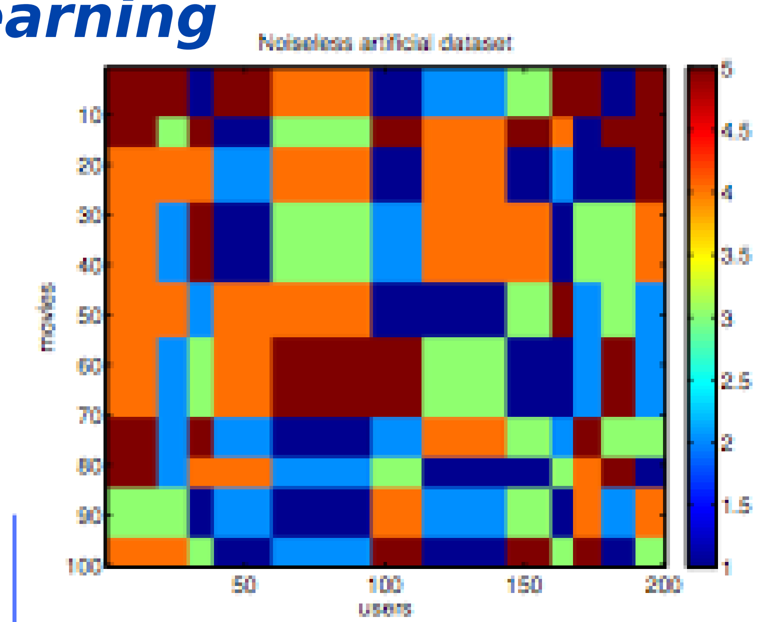


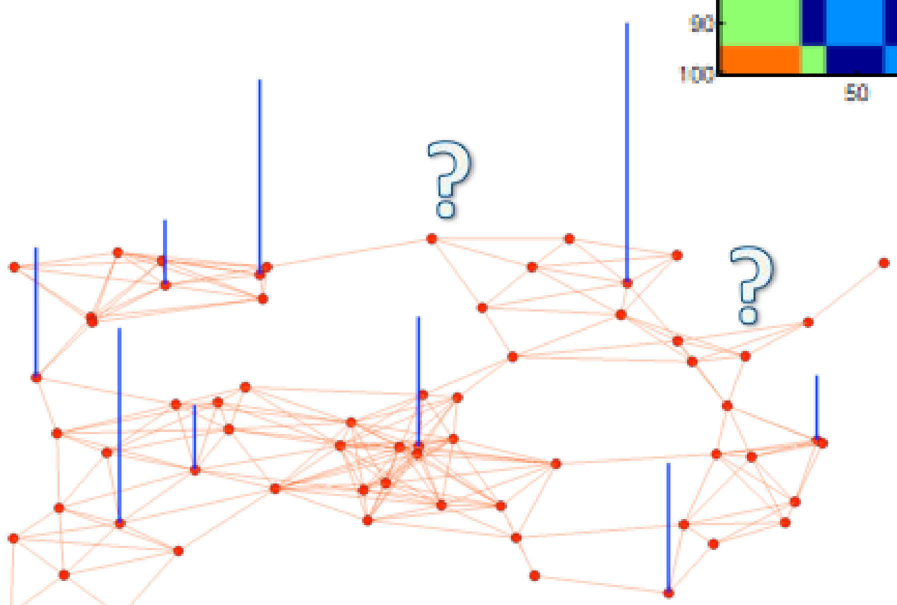
Some Typical Learning Problems

Unsupervised Learning
- Clustering
- Community detection

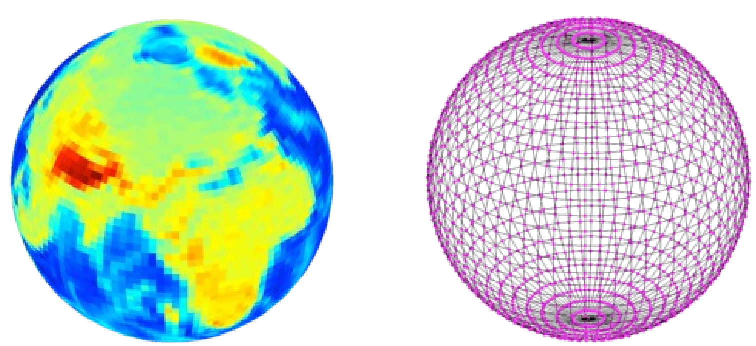


Semi-Supervised Learning
- Label propagation
- Matrix completion





Supervised Learning
- Graph convolutional NN,
« geometric Deep Learning »



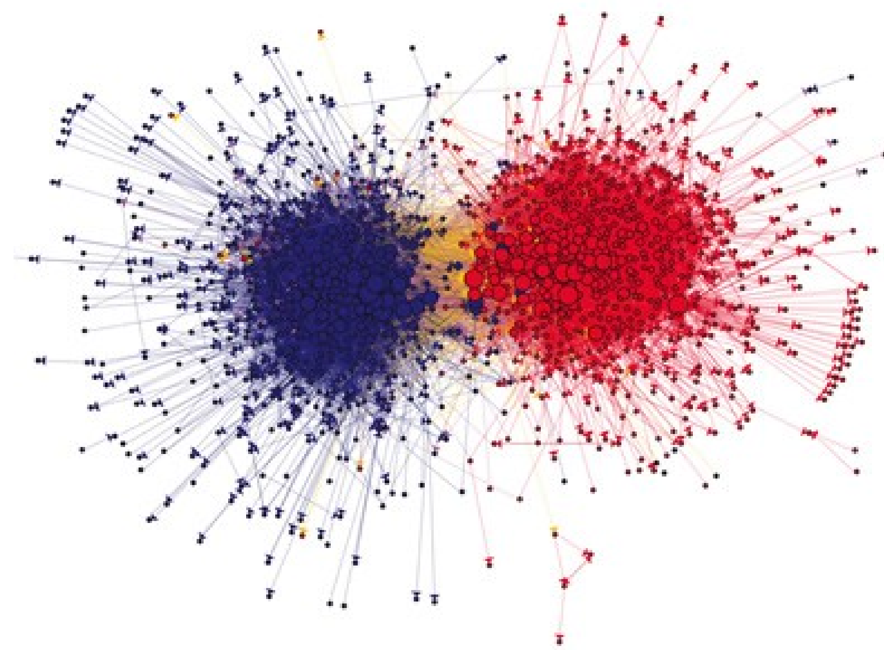
Different cases

- **Graph given (social network, brain network...) + data on the nodes,** graph information different from features information.

- **Graph computed from the data/features**. Carries the same information as the features, *is it useful*?
  yes if
  - graph helps separate the classes -> faster learning, more accurate
  - eases interpretation for humans (recommendation system)
  - carries different information : global vs local (graph constructed from global properties/ learning focused on local patterns).

Unsupervised classification

Construct a graph from the features, then :

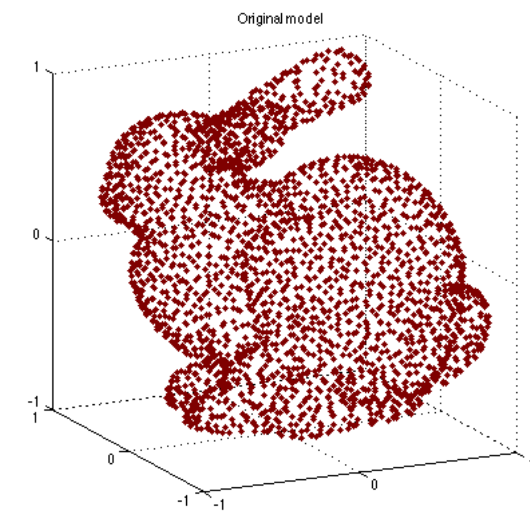


- Spectral cut, spectral clustering
  - First eigenvectors of the combinatorial or normalized graph Laplacian.
  - Fiedler vector : 2 clusters
  - k eigenvectors + k-means : k clusters
- Community detection
  - Fast and scalable, notion of modularity

[Fortunato, Community detection in graphs, 2010]

Remark on the graph design

Which distance ? How to connect ?
Popular approaches :



<u>k-NN graph</u>: regular, no hub
- Fast approximate kNN : FLANN

1) N(N-1)/2 weights to compute → k or threshold to choose, we need a sparse Laplacian

<u>Gaussian distance</u> $$w_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{4\sigma})$$

- One motivation : **graph Laplacian** converges (strongly) in probability to the **Laplace-Beltrami operator** [Belkin, Nyogi, 2005]
- Uniform distribution of points on the manifold, $n \to \infty, \quad \sigma(n) \to 0$

2) Graph : approximation of a (low-dimensional) manifold

Semi-supervised learning with / on graphs

Label propagation

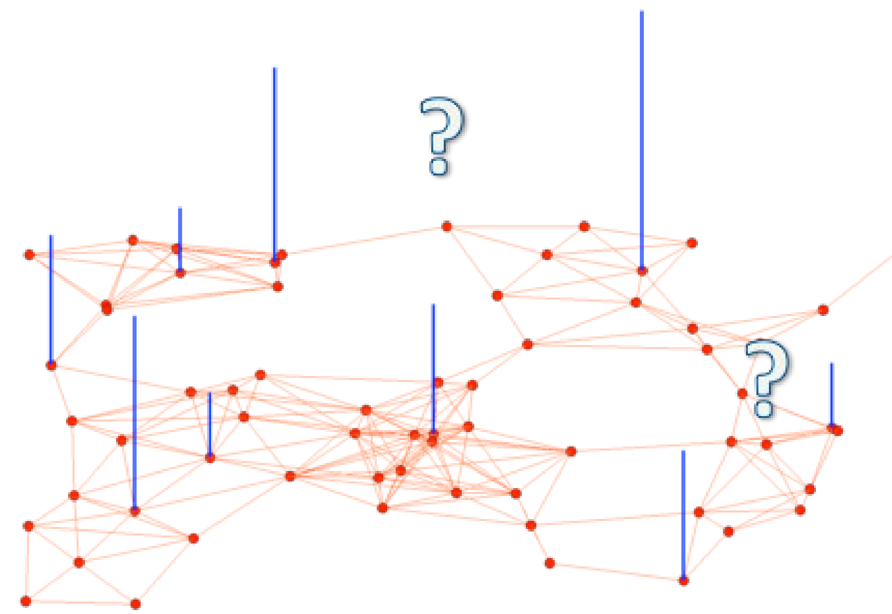
Idea : smoothness, smooth signal on a graph

$$\mathrm{Argmin}_X \|Y - AX\|_2^2 + \alpha X^T L X$$

Vector of known values

Mask

Measure of smoothness



$$\|Y - AX\|_2^2 \quad \text{or} \quad \|A \circ (Y - X)\|_2^2$$

Measure of smoothness :

$$X^T L X = \|\nabla X\|_2^2 = \sum_{i,j} w_{i,j}(x_i - x_j)^2$$

Learning with/on graphs

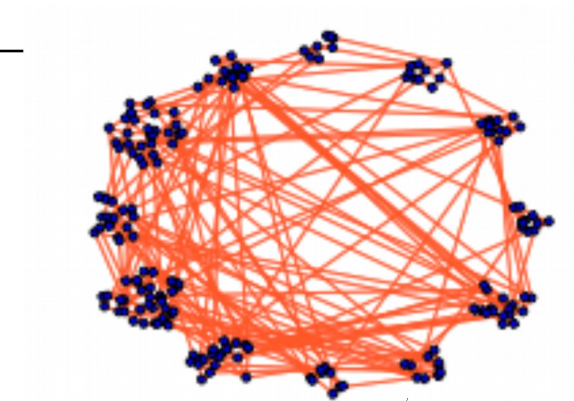
Matrix completion

$$\text{Argmin}_X \|A \circ (Y - X)\|^2 + \alpha_1 X^T L_1 X + \alpha_2 X L_2 X^T$$

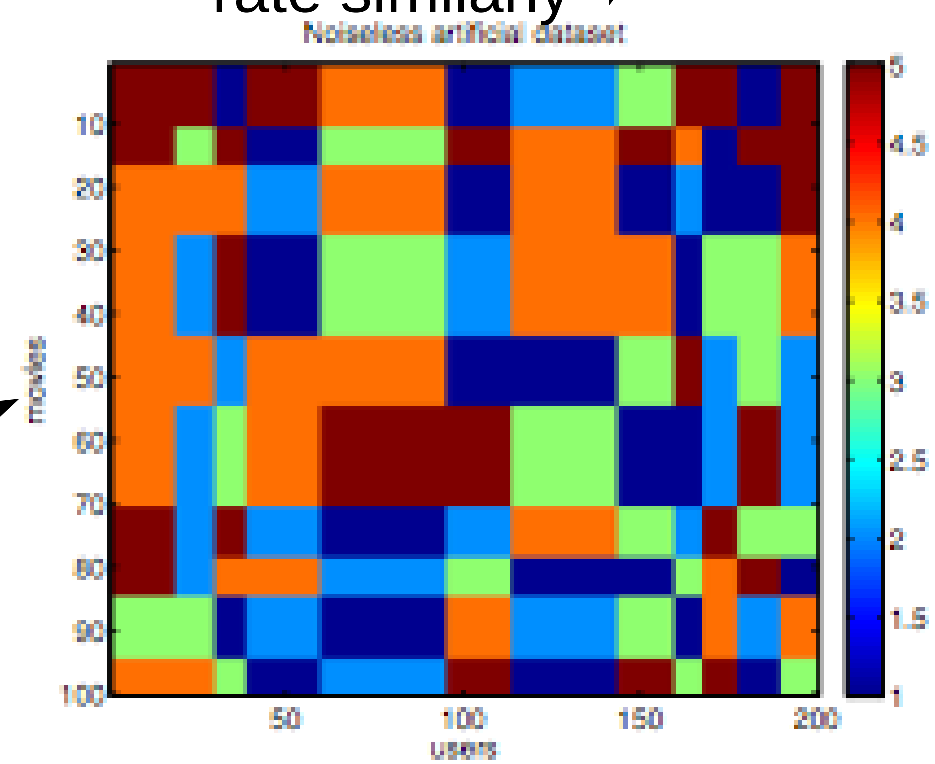
Mask

Graph of movies

Graph of users

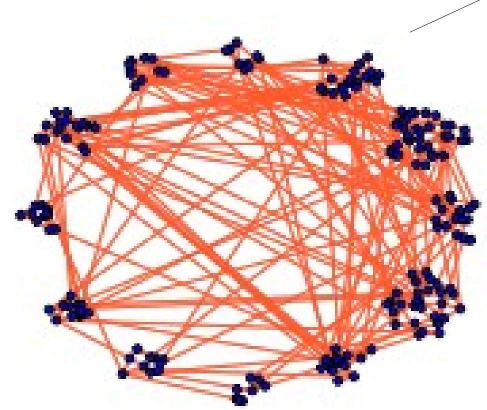


Users in communities rate similarly



$$X^T L_1 X = \sum_{i,j} w^1_{i,j} \|x_{\cdot,i} - x_{\cdot,j}\|^2$$

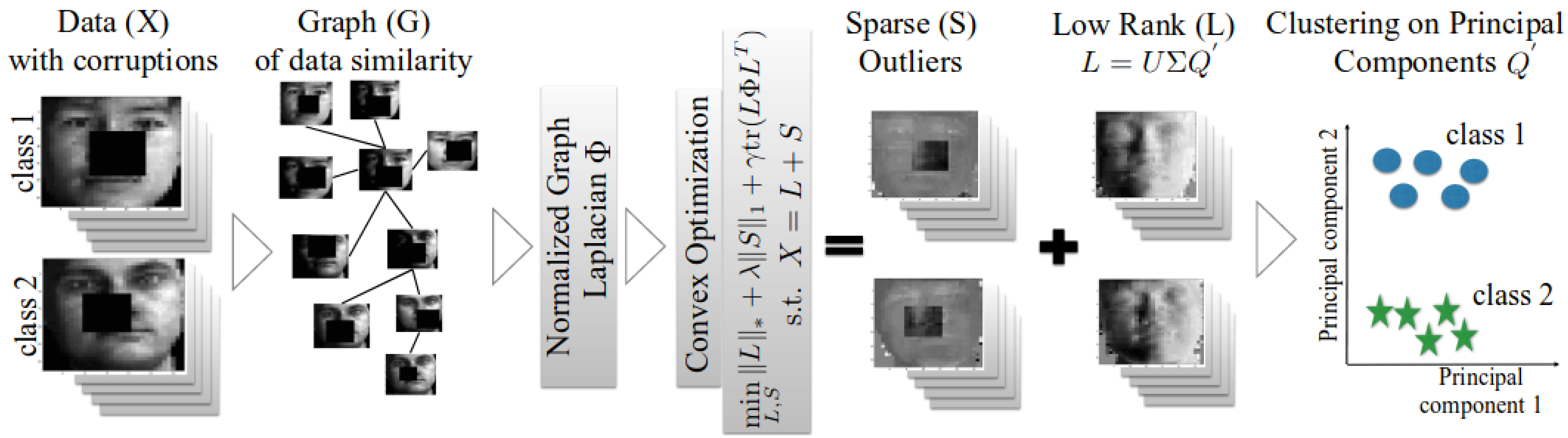
Similar movies in clusters of genres



[Kalofolias et al., Matrix Completion on Graphs, 2014]

Learning with/on graphs

Unsupervised learning : Robust PCA on graphs



Application : outliers / sparse noise

[Shahid et al., Robust Principal component analysis en graphs, 2015]

	Model	Objective	Constraints	Parameters	Graph?	Factors?	Convex?
1	PCA	$\min_{U,Q} \|X - UQ\|_F^2$	$U^TU = I$	d	no	yes	no
2	RPCA [6]	$\min_{L,S} \|L\|_* + \lambda\|S\|_1$	$X = L + S$	λ	no	no	yes
3	**PROPOSED**	$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \gamma\,\mathrm{tr}(\mathbf{L}\boldsymbol{\Phi}\mathbf{L}^T)$	$\mathbf{X} = \mathbf{L} + \mathbf{S}$	λ, γ	**YES**	**NO**	**YES**
4	GLPCA [10]	$\min_{U,Q} \|X - UQ\|_F^2 + \gamma\,\mathrm{tr}(Q\Phi Q^T)$	$QQ^T = I$				
5	RGLPCA [10]	$\min_{U,Q} \|X - UQ\|_{2,1} + \gamma\,\mathrm{tr}(Q\Phi Q^T)$		d, γ			
6	MMF [24]	$\min_{U,Q} \|X - UQ\|_F^2 + \gamma\,\mathrm{tr}(Q\Phi Q^T)$	$U^TU = I$		yes	yes	no
7	MMMF [20]	$\min_{U,Q,\boldsymbol{\alpha}} \|X - UQ\|_F^2 + \gamma\,\mathrm{tr}(Q(\sum_g \alpha_g \Phi^g)Q^T) + \beta\|\boldsymbol{\alpha}\|^2$	$U^TU = I$	d, γ, β			
8	MHMF [11]	$\min_{U,Q,\boldsymbol{\alpha}} \|X - UQ\|_F^2 + \gamma\,\mathrm{tr}(Q(\sum_g \alpha_g \Phi_h^g)Q^T) + \beta\|\boldsymbol{\alpha}\|^2$	$\mathbf{1}^T\boldsymbol{\alpha} = \mathbf{1}$				

Learning the graph

- Learn the Laplacian matrix from the signals

Space of valid Laplacians : $w_{i,j} \geq 0, \quad L^T = L, \quad \sum_i L_{i,j} = 0$

Minimization problem, smooth signals on the graph :

$$\mathrm{Argmin}_{L \in \mathcal{L}} \quad Tr(X^T L X) + \|L\|_F, \quad s.t. \quad Tr(L) = s$$

$$Tr(X^T L X) = \sum_{i,j} w_{i,j} \|x_i - x_j\|^2$$

[Dong et al., Laplacian Matrix Learning for Smooth Graph Signal Representation ICASSP 2015]

More scalable, with a log barrier on the degrees :

$$\mathrm{Argmin}_{L \in \mathcal{L}} \quad Tr(X^T L X) - \alpha \mathbf{1}^T \log(W\mathbf{1}) + \frac{\beta}{2} \|W\|_F$$

→ *No isolated node. [Kalofolias, How to learn a graph from smooth signals, AISTATS 2016]*

Limits

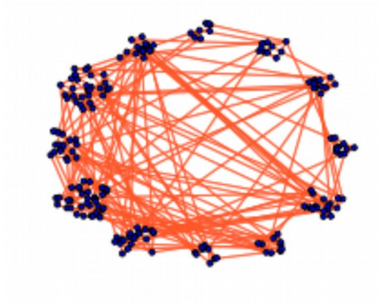
- Building the graph from the data is computationally intensive

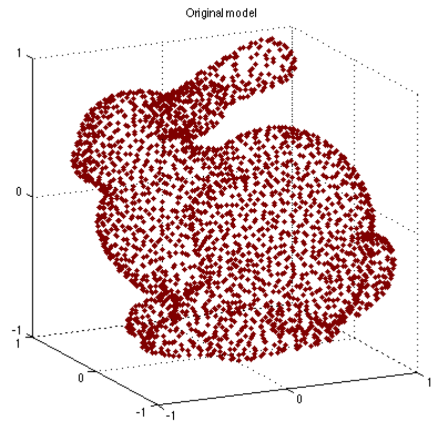
$$\frac{N(N-1)}{2}$$ Computation of distances (weights)

Alternative : Approximate nearest neighbors FLANN, $N \log N$

- Shape of the graph is important,

  - avoid hubs
  - avoid disconnected nodes
  - favor clusters ?

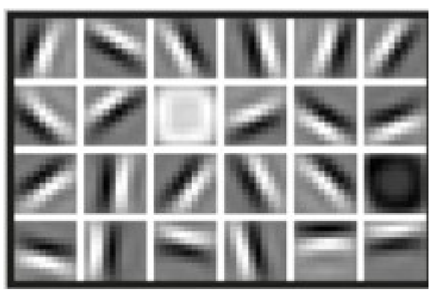


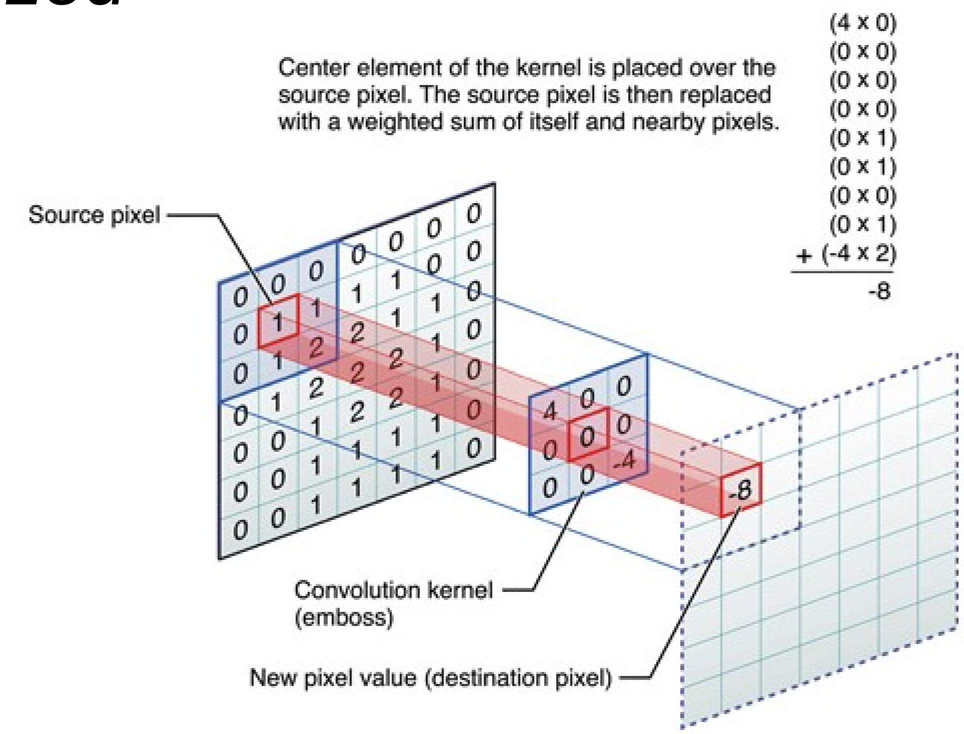


- What is a good graph ? No answer yet...

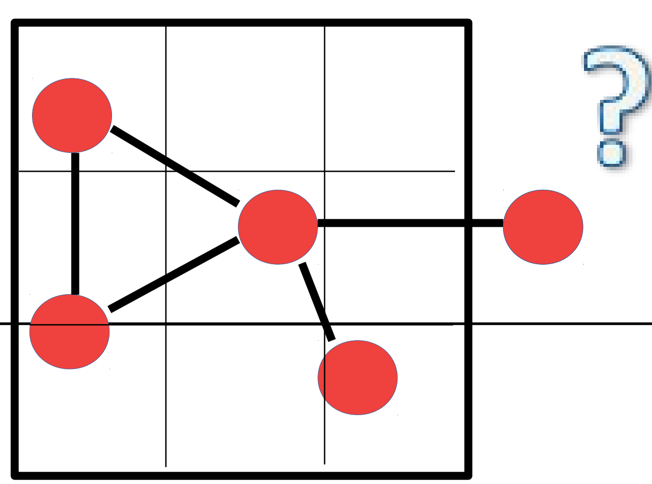
Graph CNN

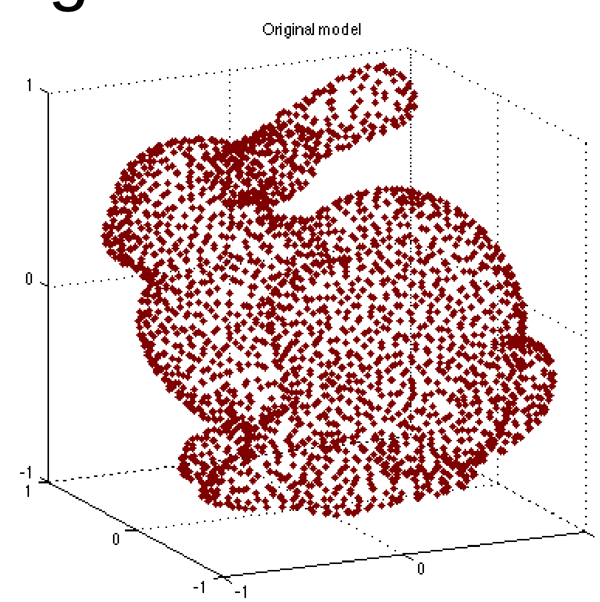
- Standard CNN : learn kernels (elementary *localized* patterns), 3x3 or 5x5 squares





- Kernel on a graph ?
- 1) Irregular and 2) the neighborhood change with position



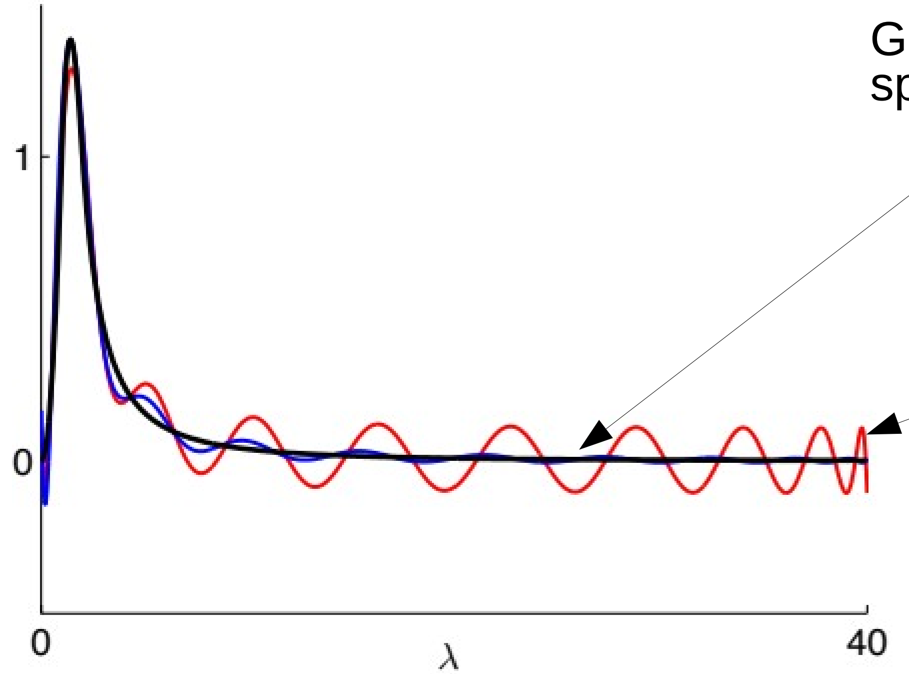


LTS EPFL

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Reminder – graph filter

- Graph filter defined on the spectral domain :



Graph filter in the spectral domain

Approximation by a Chebychev polynomial

$$g(\lambda) = \sum_{k=0}^{K-1} \theta_k \lambda^k$$

- The kernel to learn:

$$\mathcal{K}(i) = GFT^{-1} g(\lambda) GFT \delta_i = g(L)\delta_i = \sum_{k=0}^{K-1} \theta_k L^k \delta_i$$

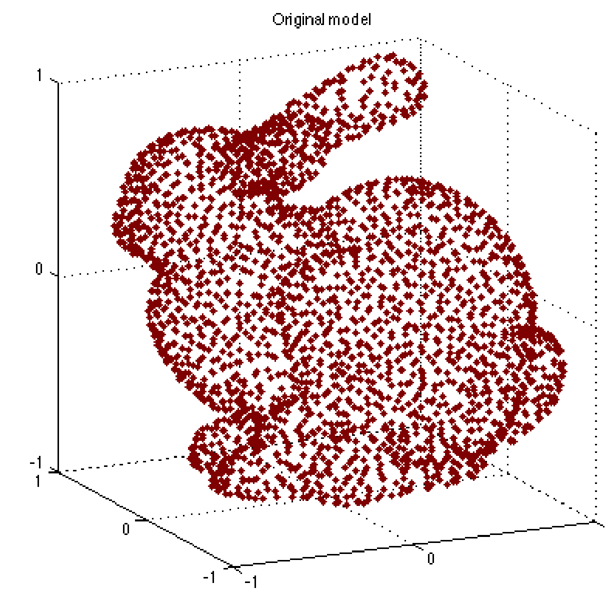
Learn the θ_k !

Graph CNN

- Graph is given

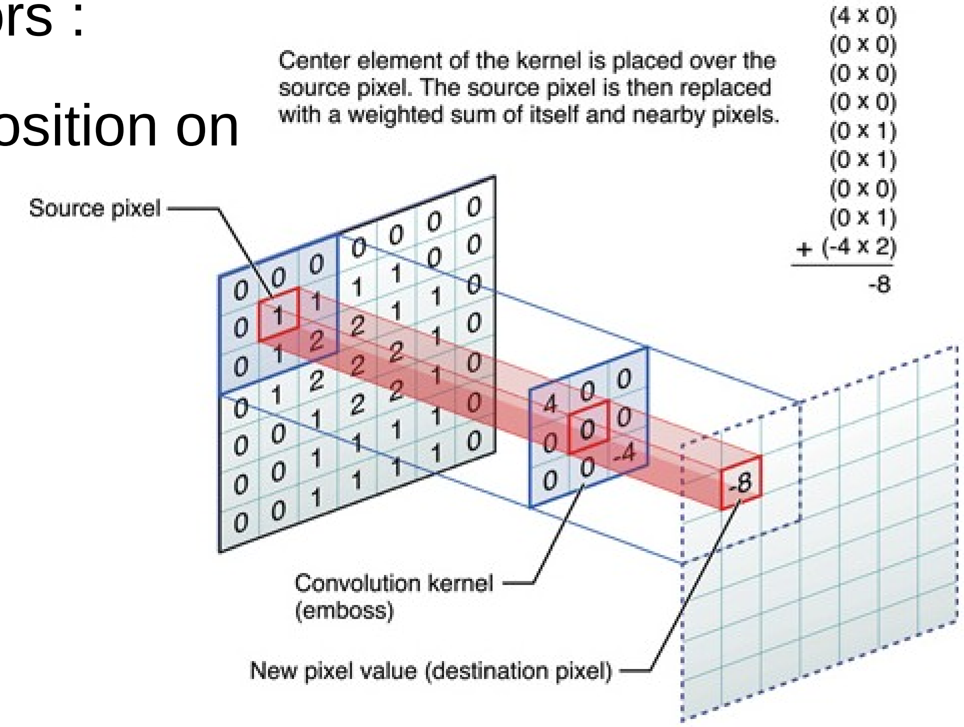
  GCNN : learn kernels defined in the spectral domai

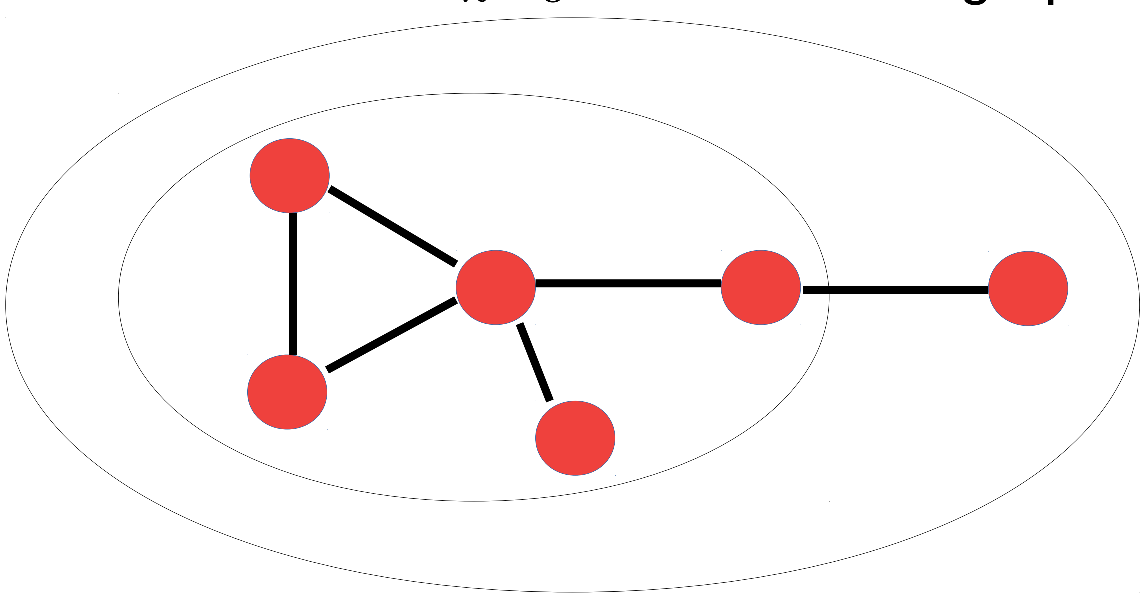
- Spectral domain « Fourier » position independent
- Learn a localized filter



$$g_\theta(L) = \sum_{k=0}^{K-1} \theta_k L^k$$

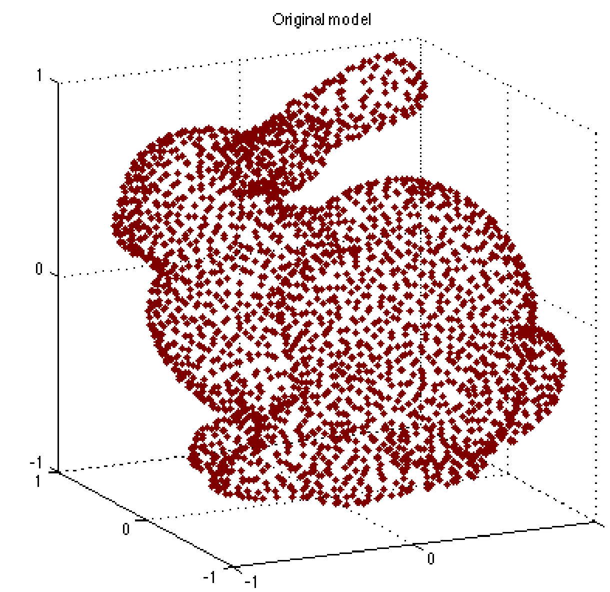
Defined on the K-hop neigbors :
localized
Thetas independent of the position on
the graph

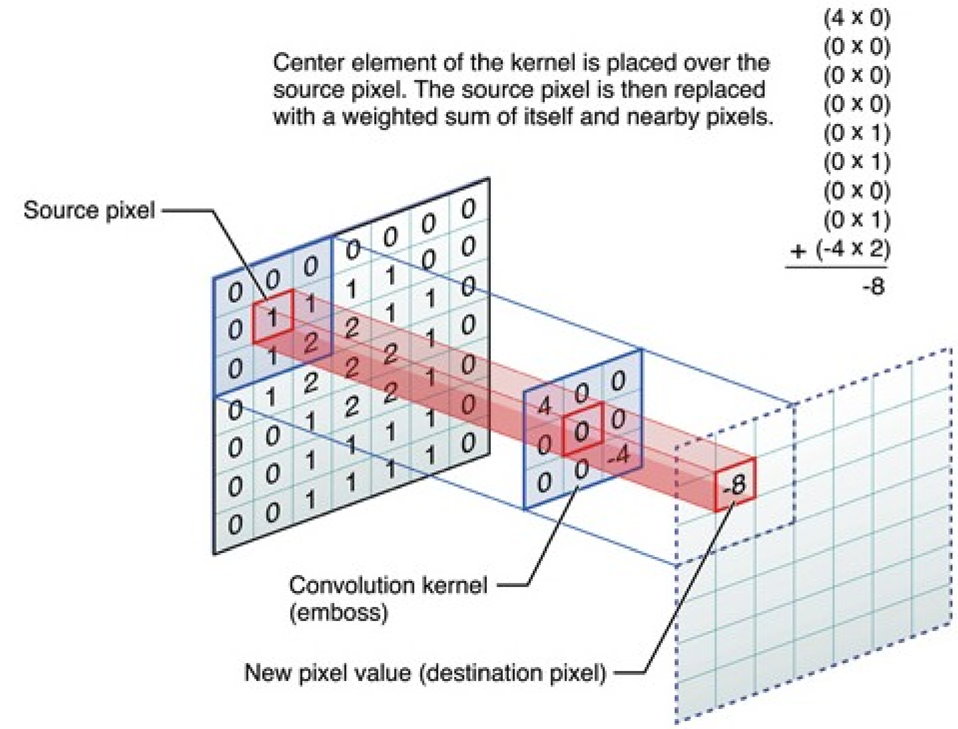


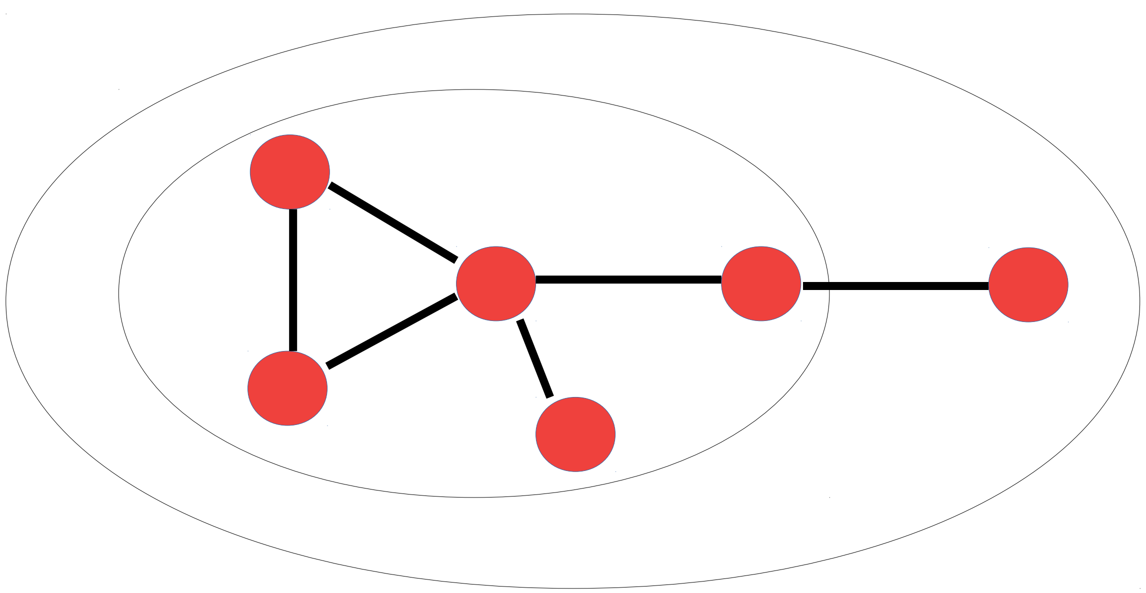


Graph CNN

- Pooling : graph coarsening
- Any coarsening method may be used provided it is fast and parallel







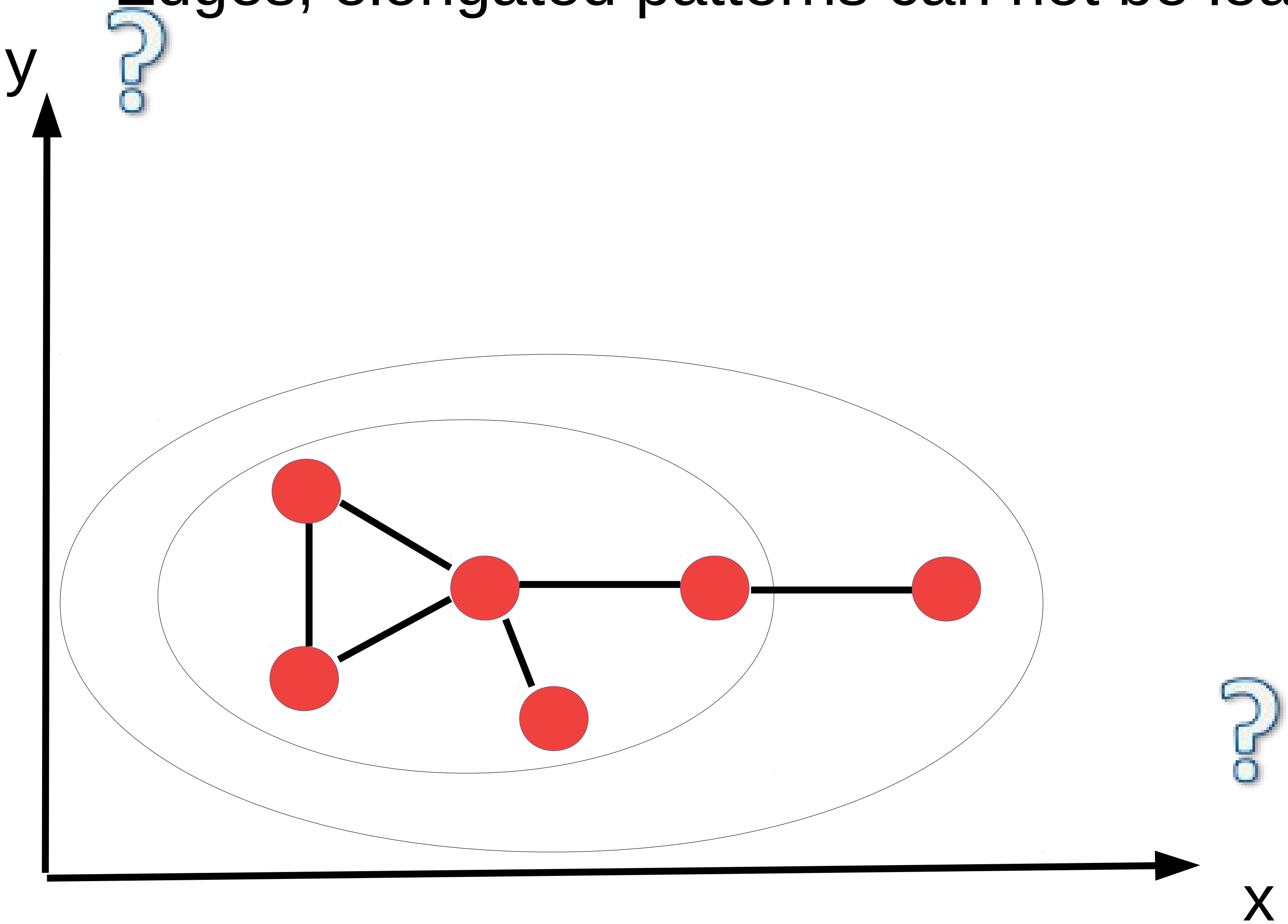
Graph CNN limitations

- Graphs do not have directions

  Kernels are isotropic

$$g_\theta(L) = \sum_{k=0}^{K-1} \theta_k L^k$$

- Edges, elongated patterns can not be learned



0	0	0
3	5	3
0	0	0