



Electronique

Energie Electrique

Automatique

Master 2 3EA, Parcours RoVA



# Systemes Robotiques Hétérogènes et Coopératifs

UPJV, Département EEA

**Fabio MORBIDI**

Laboratoire MIS

Équipe Perception Robotique

E-mail : [fabio.morbidi@u-picardie.fr](mailto:fabio.morbidi@u-picardie.fr)

CM, TD : Mardi 15h30-18h30 et Mercredi 13h30-16h30, salle CURI 305

TP : Mercredi 13h30-16h30, salle TP204

# Plan du cours

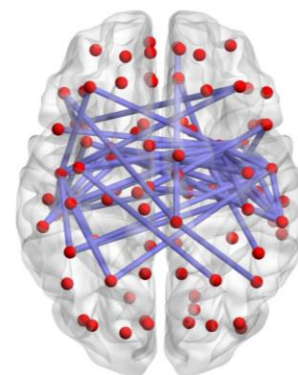
1. Introduction aux systèmes multi-agents
2. Théorie des graphes
3. Systèmes dynamiques connectés en réseaux : protocole de consensus
4. Traitement du signal sur graphes



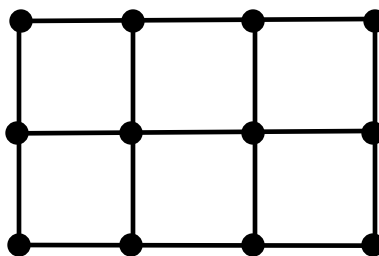
# Signaux sur graphes : généralités

Aujourd'hui, de plus en plus de données structurées sur **forme de graphe** :

- Réseaux sociaux
- Réseaux de régulation (génique)
- Réseaux cérébraux fonctionnels
- Graphes des citations
- Réseaux électriques
- Réseaux de capteurs
- Réseaux de chaleur et d'égouts
- Réseaux routiers



Domaine d'un signal discret 1D (segment)



Domaine d'un signal discret 2D (grille régulière)



Domaine irrégulier (graphe)



# Signaux sur graphes : définition

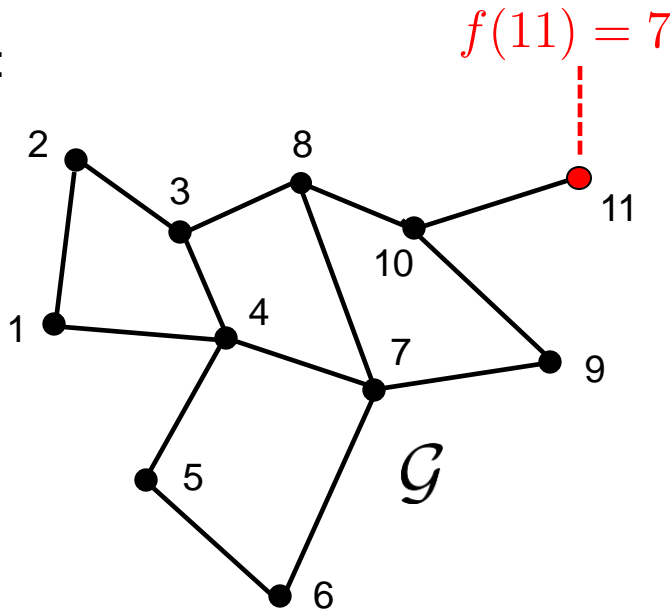
## Définition :

Étant donné un graphe non orienté  $\mathcal{G} = (V, E)$ , un *signal sur graphe* est une fonction  $f(x)$  à valeurs réelles définie sur les sommets du graphe  $\mathcal{G}$ , c'est-à-dire :

$$f(x) : V \mapsto \mathbb{R}$$

On écrira  $\mathbf{f} = [f(1), f(2), \dots, f(n)]^T$ , pour indiquer le vecteur qui contient les  $n$  valeurs prises par le signal  $f(x)$  sur les sommets du graphe  $\mathcal{G}$

## Exemple 1 :



$$\mathbf{f} = \begin{bmatrix} f(1) \\ f(2) \\ \vdots \\ f(11) \end{bmatrix} = \begin{bmatrix} -3 \\ 4 \\ \vdots \\ 7 \end{bmatrix}$$



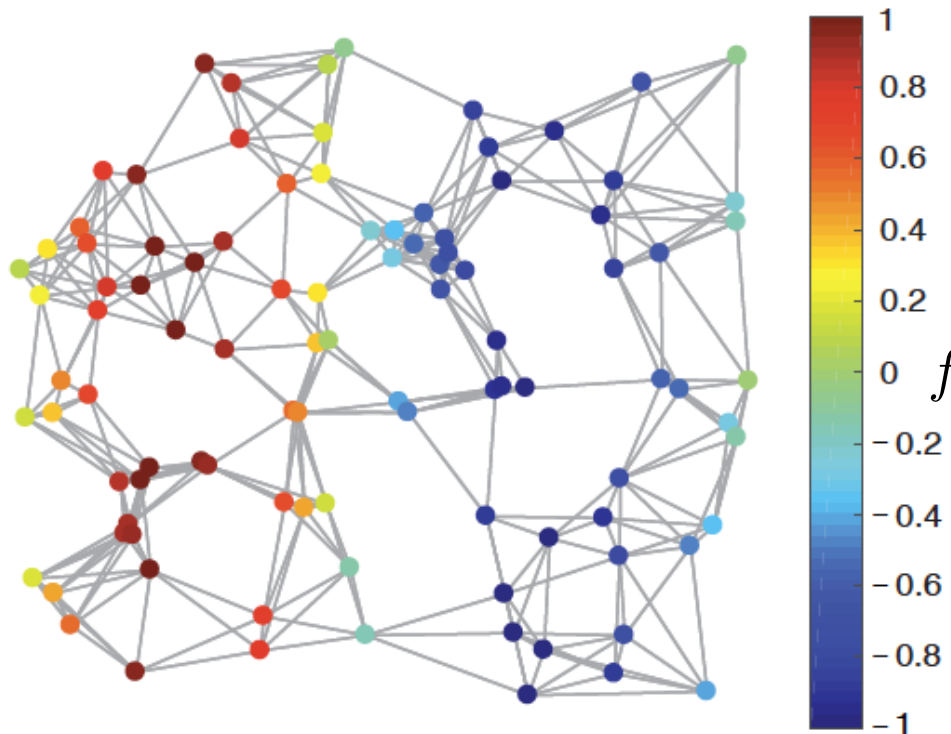
# Signaux sur graphes : exemples

Exemple 2 :

Signal

$$f(i) = \sin(2\pi i/n), i \in \{1, 2, \dots, n\}$$

sur un graphe aléatoire 2D avec  $n = 100$  sommets

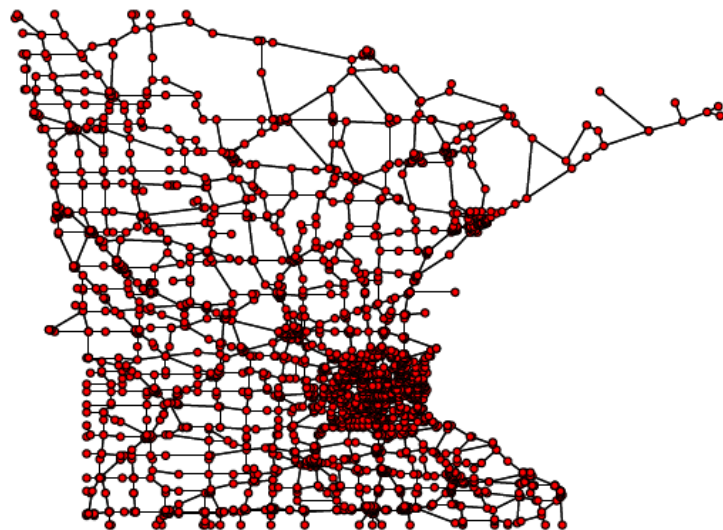


# Signaux sur graphes : exemples

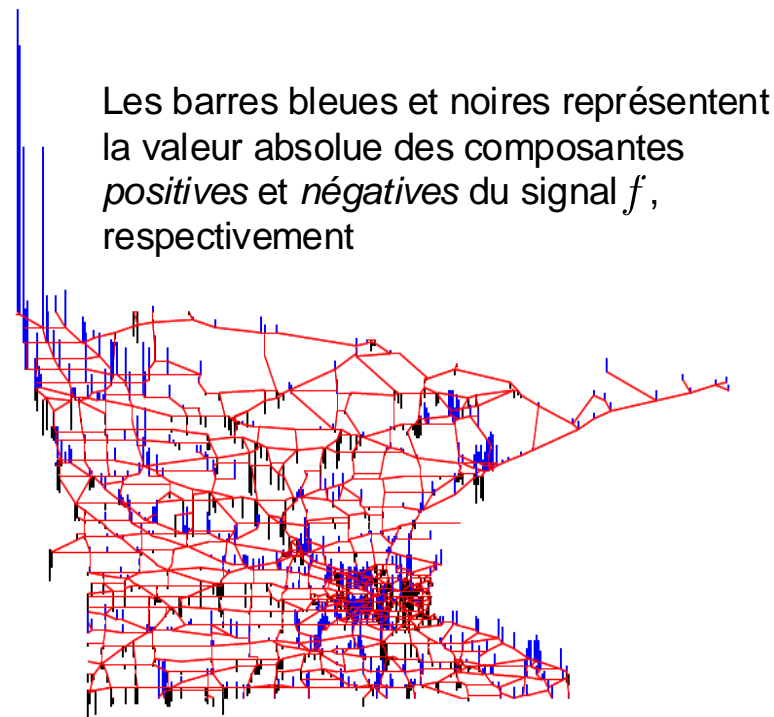
## Exemple 3 :

Réseau routier de l'état du Minnesota (2600 sommets et 3300 arêtes)

<https://networkrepository.com/road-minnesota.php>



Réseau routier



Les barres bleues et noires représentent la valeur absolue des composantes positives et négatives du signal  $f$ , respectivement

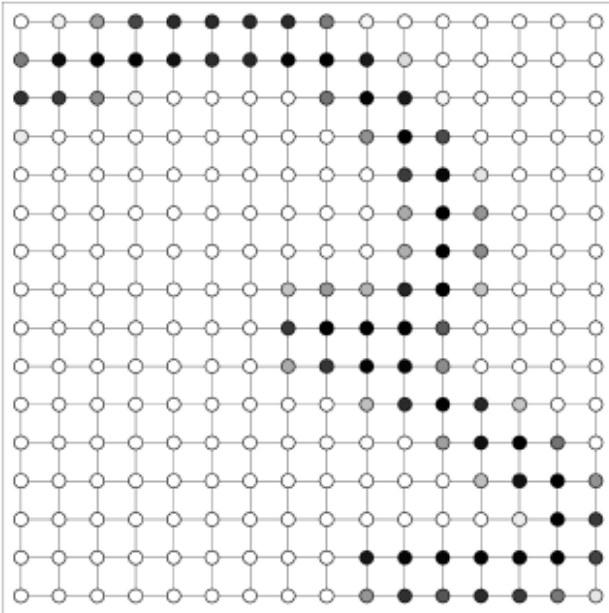
Signal sur le réseau

- “Vertex-frequency analysis on graphs”, D.I. Shuman, B. Ricaud, P. Vandergheynst, Appl. Comput. Harmon. Anal., vol. 40, n. 2, pp. 260-291, 2016



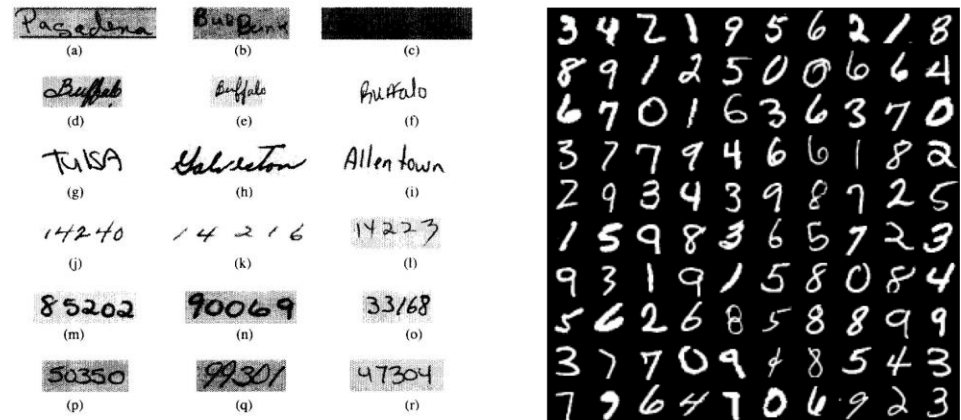
# Signaux sur graphes : exemples

## Exemple 4 (Pixels d'une image) :



Graphe grille pour représenter l'image d'un numéro (le « 3 ») écrit à la main (dataset USPS - U.S. Postal Service)

- Graphe grille pour représenter une image
- Chaque sommet du graphe correspond à un pixel de l'image. Le signal associé à chaque sommet est l'intensité du pixel correspondant (niveau de gris: [0, 255])
- Dans un graphe qui est *indépendant* du signal, tous les poids sur les arêtes ont la même valeur (la valeur 1)



- "A database for handwritten text recognition research", J.J. Hull, IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, n. 5, pp. 550-554, 1994



# Traitement du signal sur graphes

## Définition :

Le *traitement du signal sur graphes* (« *graph signal processing* ») est l'étude des outils de calcul pour traiter et analyser les données définies sur des domaines ou des structures de corrélation non régulières (graphes)

- Dans ce qui suit, nous verrons quelque notion de base :
  - Transformée de Fourier d'un signal sur graphe
  - Produit de convolution de deux signaux sur graphe
  - Filtrage de signaux sur graphe
  - Échantillonnage de signaux sur graphe
  - Réseaux de neurones sur graphes (GNN)

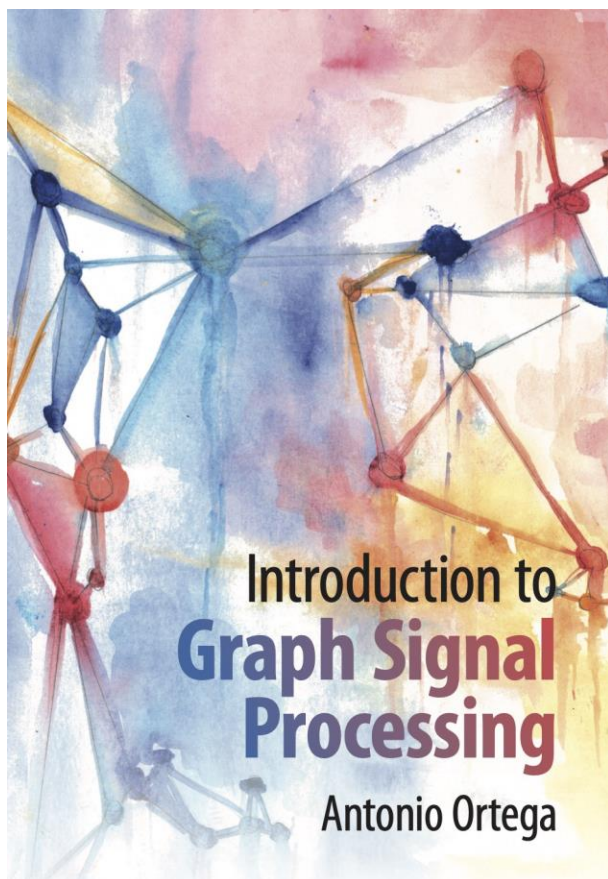
“*Graph Signal Processing: Overview, Challenges, and Applications*“, A. Ortega, P. Frossard, J. Kovačević, J.M.F. Moura, P. Vandergheynst, Proc. IEEE, vol. 106, n. 5, pp. 808-828, 2018





# Traitement du signal sur graphes

- Domaine de recherche très récent (2010 – aujourd’hui)



◀ Premier livre publié en 2022 :

*“Introduction to Graph Signal Processing”*

A. Ortega, Cambridge University Press, 2022

[www.graph-signal-processing-book.org](http://www.graph-signal-processing-book.org)

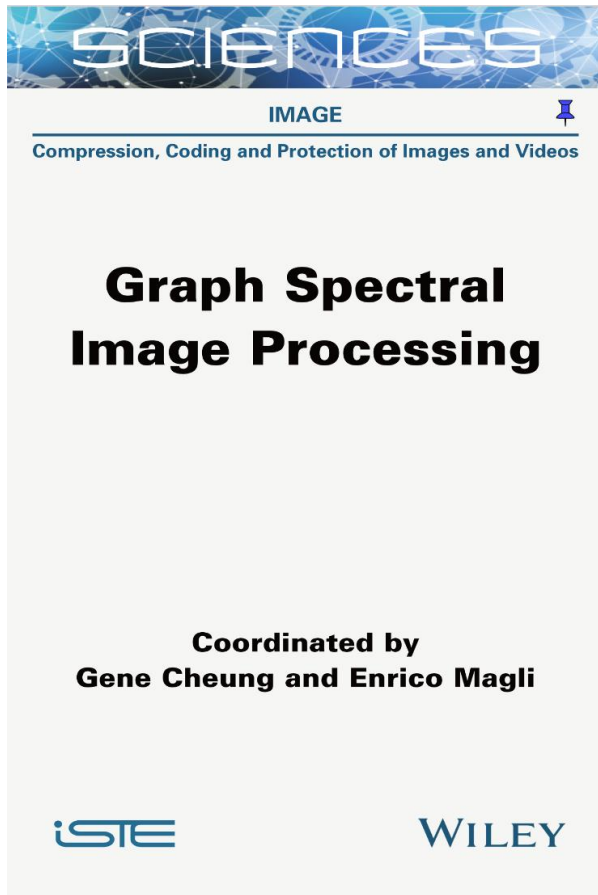
État de l’art récent :

- *“Discrete signal processing on graphs”*,  
A. Sandryhaila, J.M.F. Moura, IEEE Trans.  
Signal Process., vol. 61, n. 7, pp. 1644-1656, 2013
- *“The emerging field of signal processing on graphs:  
Extending high-dimensional data analysis to networks  
and other irregular domains”*, D. I. Shuman, S.K. Narang,  
P. Frossard, A. Ortega, P. Vandergheynst, IEEE Signal  
Process. Mag., vol. 30, n. 3, pp. 83-98, 2013
- *“Cooperative and Graph Signal Processing: Principles  
and Applications”*, P.M. Djuric, C. Richard (Édit.),  
Academic Press, 2018



# Traitement du signal sur graphes

- Domaine de recherche très récent (2010 – aujourd’hui)



◀ Pour le traitement d’images :

“*Graph Spectral Image Processing*”  
G. Cheung, E. Magli (Édit.), ISTE Wiley,  
Novembre 2021

[www.iste.co.uk/book.php?id=1790](http://www.iste.co.uk/book.php?id=1790)

État de l’art récent :

- “*Graph Spectral Image Processing*”, G. Cheung, E. Magli, Y. Tanaka, M.K. Ng, Proc. IEEE, vol. 106, n. 5, pp. 907-930, 2018



# Transformée de Fourier sur graphe

Nous avons vu que la matrice laplacienne  $\mathbf{L}$  d'un graphe non orienté est une matrice *symétrique*. Elle admet donc la factorisation spectrale suivante :

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$$

avec

$\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  : matrice diagonale  $n \times n$  avec les valeurs propres de  $\mathbf{L}$  en ordre croissant sur la diagonale principale

$\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n]$  : matrice orthogonale  $n \times n$  dont les colonnes sont les vecteurs propres normalisés de  $\mathbf{L}$

## Définition (GFT)

La *transformée de Fourier sur graphe* (GFT)  $\mathcal{F}[\cdot]$  du signal  $f(x)$  est définie comme suit :

$$\mathcal{F}[f] = \hat{f} \triangleq \mathbf{U}^T f$$

La GFT  $\hat{f}$  d'un signal  $f$  défini sur les sommets d'un graphe  $\mathcal{G}$  est donc l'expansion de  $f$  selon les vecteurs propres de  $\mathbf{L}$



# Transformée de Fourier sur graphe

## Définition (IGFT)

De la même façon, la *transformée de Fourier inverse sur graphe* (IGFT) du signal  $\hat{f}$  s'écrit, en matriciel :

$$\mathcal{F}^{-1}[\hat{f}] \triangleq \mathbf{U} \hat{f}$$

## Remarques :

- La GFT *projette* le signal sur graphe d'entrée sur l'espace engendré par les colonnes orthogonales de  $\mathbf{U}$  (appelée aussi *base de Fourier*). Les éléments du signal  $\hat{f}$  transformé, sont les coordonnées du signal sur graphe dans un nouvel espace, de manière à ce que le signal d'entrée puisse être représenté comme :

$$\mathbf{f} = \sum_{i=1}^n \hat{f}(i) \mathbf{u}_i$$

ce que est exactement l'IGFT

- Il est facile à vérifier que la GFT (comme la transformée de Fourier classique) satisfait l'*égalité de Parseval*, c'est-à-dire :

$$\|\mathbf{f}\|^2 = \|\hat{f}\|^2$$



# Transformée de Fourier sur graphe

## Remarques (suite)

- Comme la transformée de Fourier classique, la GFT permet de représenter un signal dans deux domaines différents : le *domaine des sommets* et le *domaine spectral du graphe* (ou *domaine de Fourier sur graphe*)
- On peut interpréter les valeurs propres de  $\mathbf{L}$  comme des *fréquences* : une valeur propre élevée entraîne une plus grande variabilité du vecteur propre correspondant
- Il est à noter que la définition de GFT et d'IGFT dépend du choix des vecteurs propres de  $\mathbf{L}$ , qui n'est pas nécessairement *unique*
- En alternative, une GFT peut être définie en utilisant la *matrice d'adjacence* du graphe  $\mathbf{A}$  et cette définition est aussi utilisable avec les graphes orientés\*. D'ailleurs, on peut interpréter  $\mathbf{A}$  et  $\mathbf{L}$  (et leurs *formes symétriques normalisées*,  $\mathbf{A}^{\text{sym}} = \mathbf{\Delta}^{-1/2} \mathbf{A} \mathbf{\Delta}^{-1/2}$  et  $\mathbf{L}^{\text{sym}} = \mathbf{\Delta}^{-1/2} \mathbf{L} \mathbf{\Delta}^{-1/2}$ , respectivement) comme une classe générale de *opérateurs de décalage des graphes* (GSO: « graph shift operators »)

\* "Discrete signal processing on graphs", A. Sandryhaila, J.M.F. Moura, IEEE Trans. Signal Process., vol. 61, n. 7, pp. 1644-1656, 2013



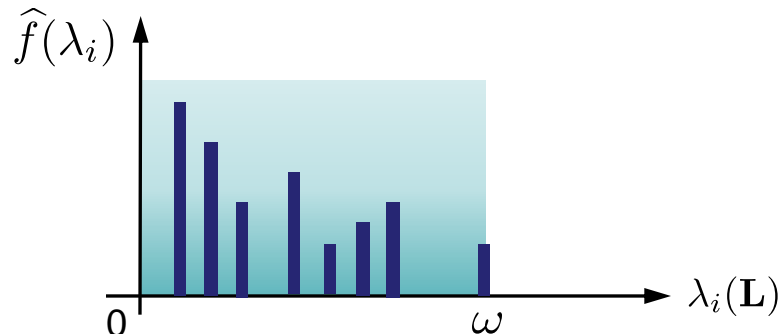
# Transformée de Fourier sur graphe

## Remarques (suite)

- Sur la base de l'interprétation des valeurs propres de  $\mathbf{L}$  comme des *fréquences*, on peut introduire la définition suivante :

## Définition (Signal sur graphe à bande limitée)

Un signal sur graphe avec une *bande limitée*  $\omega$ , est un signal qui a zéro coefficients de la GFT au dessus de sa bande passante  $\omega$ , à savoir son support spectral est restreint à l'ensemble des fréquences  $[0, \omega]$

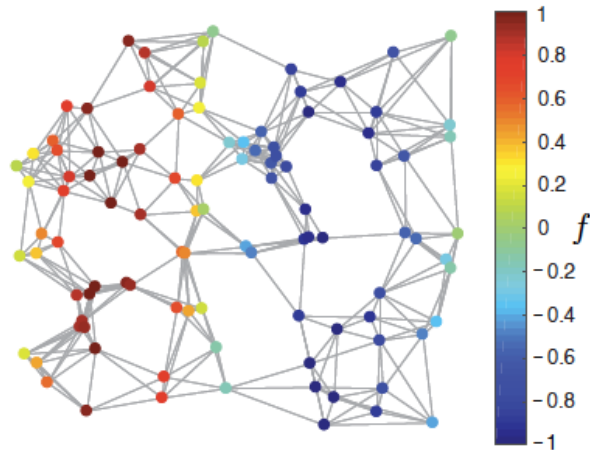


- L'espace de tous les signaux à bande limitée  $\omega$  sur un graphe  $\mathcal{G}$  s'appelle *espace de Paley-Wiener*, noté  $\text{PW}_\omega(\mathcal{G}) \subset \mathbb{R}^n$ .



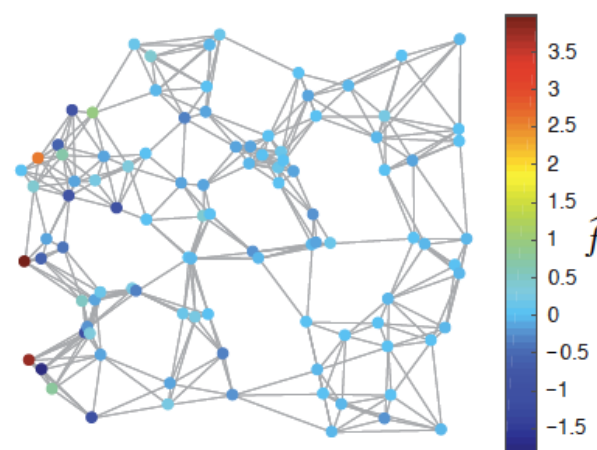
# Transformée de Fourier sur graphe

Exemple 1 :

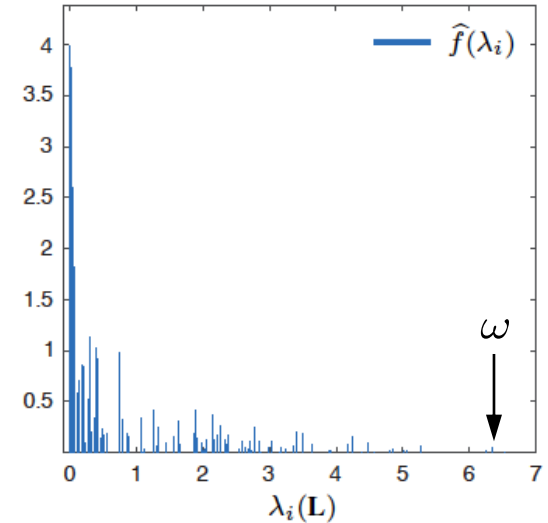


Signal défini sur un graphe  
aléatoire 2D avec 100 sommets:

$$f(i) = \sin(2\pi i/n), i \in \{1, 2, \dots, n\}$$



GFT de  $f$



Signal  $f$  dans le  
domaine spectral  
du graphe



# Transformée de Fourier sur graphe

Exemple 2 (Signal lisse sur graphe) :

- Pour mesurer la *régularité* d'un signal  $\mathbf{f}$  sur un graphe  $\mathcal{G}$ , on peut utiliser la *forme quadratique laplacienne* (cf. le potentiel laplacien) définie par :

$$Q(\mathbf{L}) = 2 \Phi_{\mathcal{G}}(\mathbf{f}) = \mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i,j=1}^n a_{ij} (\mathbf{f}(i) - \mathbf{f}(j))^2$$

où  $a_{ij}$  est l'élément  $(i, j)$  de la matrice d'adjacence  $\mathbf{A}$  du graphe  $\mathcal{G}$

- Clairement,  $Q(\mathbf{L}) = 0$  si  $\mathbf{f}$  est un *signal constant* sur le graphe  $\mathcal{G}$
- De manière plus générale, on peut remarquer que plus la valeur de  $Q(\mathbf{L})$  est *faible*, plus les valeurs des signaux sur les sommets voisins sont *similaires* (c'est-à-dire, plus la variation de  $\mathbf{f}$  par rapport à  $\mathcal{G}$  est faible)

- “*Learning Graphs from Data: A signal representation perspective*”, X. Dong, D. Thanou, M. Rabbat, P. Frossard, IEEE Signal Processing Magazine, vol. 36, n. 3, pp. 44-63, 2019

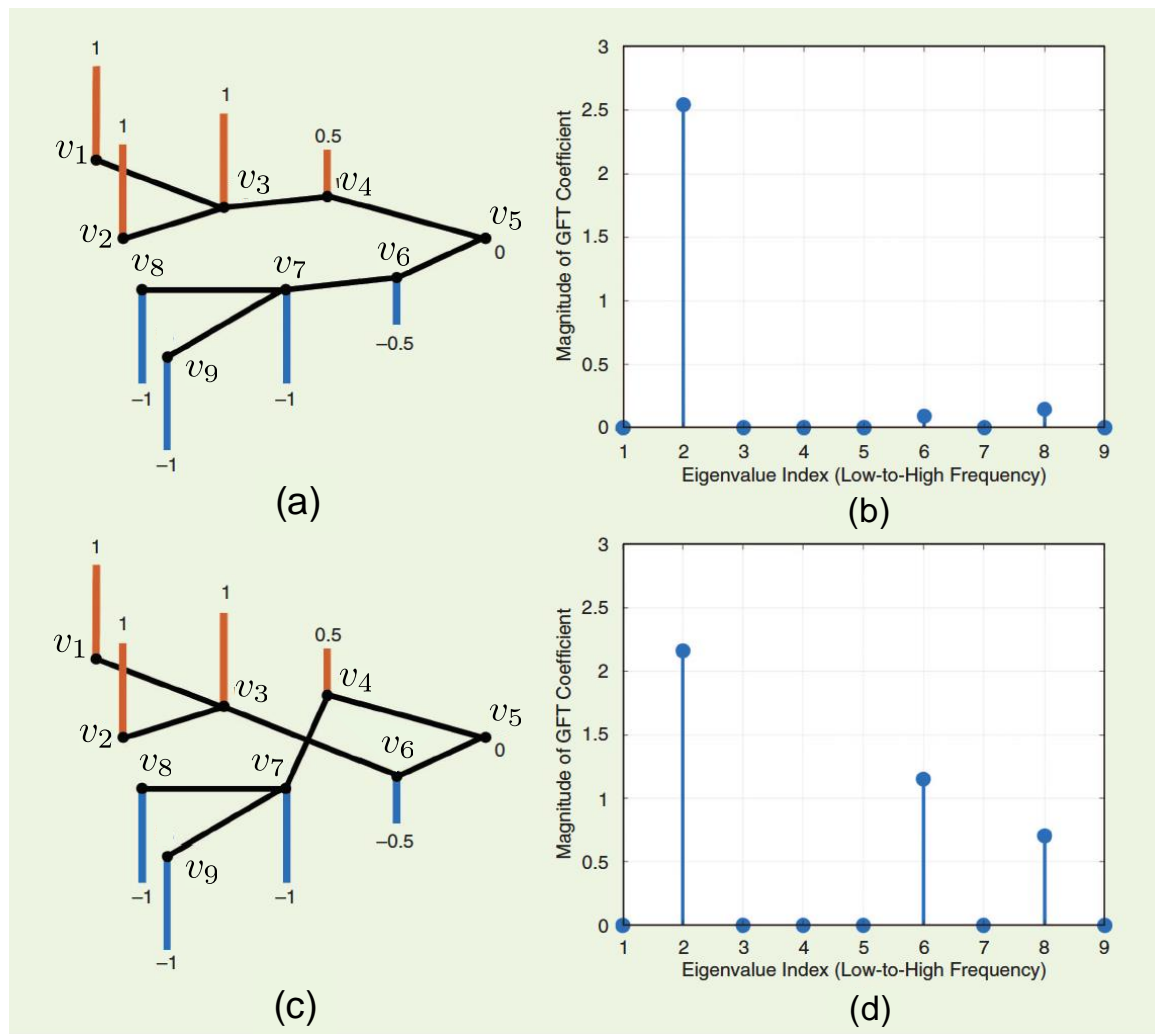




# Transformée de Fourier sur graphe

## Exemple 2 (Signal lisse sur graphe) :

- (a) Un signal lisse sur graphe avec  $Q(\mathbf{L}) = 1$
- (b) Les coefficients de sa GFT dans le domaine spectral du graphe. Le signal a une représentation lisse sur le graphe, car ses valeurs varient lentement sur les arêtes du graphe. Le signal consiste principalement de composantes à *basse fréquence* dans le domaine spectral du graphe
- (c) Un signal moins lisse sur graphe avec  $Q(\mathbf{L}) = 5$
- (d) Les coefficients de sa GFT dans le domaine spectral du graphe. Un choix *différent* du graphe produit une représentation *différente* du même signal



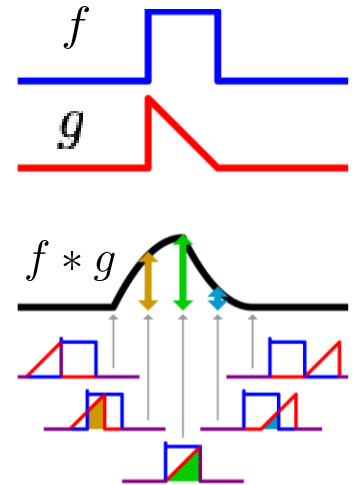
# Produit de convolution

## Rappel :

Le **produit de convolution** de deux signaux  $f$  et  $g$ , est un autre signal noté «  $f * g$  », défini par :

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

- Même si on utilise le symbole  $t$ , il ne représente pas nécessairement un temps
- A chaque instant de temps  $t$ , la formule de convolution nous donne l'*aire* sous le signal  $f(\tau)$  pondérée par le signal  $g(-\tau)$  décalé d'une quantité  $t$ . Comme  $t$  varie, le signal de pondération  $g(t - \tau)$  « met en évidence » différentes parties du signal d'entrée  $f(\tau)$



La **transformée de Fourier** d'un produit de convolution s'obtient par multiplication des transformées de Fourier des signaux :

- Si  $f$  et  $g$  sont intégrables alors :

$$\mathcal{F}[f * g] = \mathcal{F}[f] \mathcal{F}[g]$$

- Si  $f$  et  $g$  sont de carré intégrable alors :

$$f * g = \mathcal{F}^{-1} [\mathcal{F}[f] \mathcal{F}[g]]$$

où  $\mathcal{F}$  désigne la transformée de Fourier et  $\mathcal{F}^{-1}$  la transformée de Fourier inverse



# Produit de convolution sur graphe

## Problème :

- La définition de convolution entre deux signaux *ne peut pas être appliquée directement* aux signaux sur graphes, car l'*opération de translation* n'est pas définie pour ces signaux
- Cependant, on peut remplacer la translation par exponentiel complexe de la transformée de Fourier classique ( $\forall t_0$  si  $h(t) = f(t - t_0)$  alors  $\mathcal{F}[h] = e^{-2\pi i t_0 \xi} \mathcal{F}[f]$ ), avec les vecteurs propres de la matrice laplacienne et introduire un *opérateur de convolution généralisé*

## Définition (Produit de Hadamard)

Le *produit de Hadamard* (ou « produit élément par élément »)  $\mathbf{A} \circ \mathbf{B}$  de deux matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ , est une matrice  $m \times n$  avec éléments donnés par :

$$[\mathbf{A} \circ \mathbf{B}]_{ij} = [\mathbf{A}]_{ij} [\mathbf{B}]_{ij}$$

## Exemple :

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & -5 \\ 0 & 2 & 7 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 4 & -2 & 1 \\ 1 & -1 & 6 \end{bmatrix}$$

$$\mathbf{A} \circ \mathbf{B} = \begin{bmatrix} 4 & -6 & -5 \\ 0 & -2 & 42 \end{bmatrix}$$

Commande Matlab : `A .* B`



# Produit de convolution sur graphe

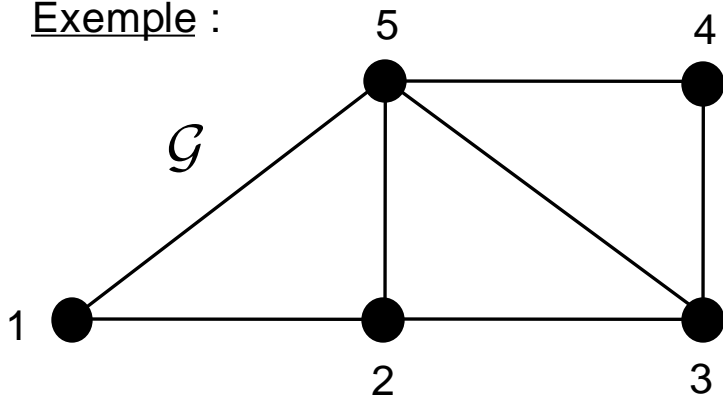
**Définition** (*Convolution de deux signaux sur graphes*)

Soient  $f(x)$  et  $g(x)$  deux signaux définis sur le même graphe  $\mathcal{G}$ , et  $\mathbf{f}$  et  $\mathbf{g}$  les vecteurs contenant les valeurs prises par les deux signaux sur les sommets de  $\mathcal{G}$ .

On définit alors la *convolution sur graphe* “ $*_{\mathcal{G}}$ ” de  $f(x)$  et  $g(x)$  de la façon suivante :

$$\mathbf{f} *_{\mathcal{G}} \mathbf{g} \triangleq \mathcal{F}^{-1} [\mathcal{F}[\mathbf{f}] \circ \mathcal{F}[\mathbf{g}]] = \mathbf{U}(\mathbf{U}^T \mathbf{f} \circ \mathbf{U}^T \mathbf{g})$$

Exemple :



$$\mathbf{f} = [1, 2, 4, 1, -3]^T$$

$$\mathbf{g} = [-1, 0, 5, 6, 1]^T$$

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 0.4472 & 0.6533 & 0.5000 & -0.2706 & -0.2236 \\ 0.4472 & 0.2706 & -0.5000 & 0.6533 & -0.2236 \\ 0.4472 & -0.2706 & -0.5000 & -0.6533 & -0.2236 \\ 0.4472 & -0.6533 & 0.5000 & 0.2706 & -0.2236 \\ 0.4472 & 0.0000 & -0.5000 & -0.0000 & 0.8944 \end{bmatrix}$$

$$\mathbf{f} *_{\mathcal{G}} \mathbf{g} = [5.1877, 5.6168, 1.5386, 1.9677, 10.2859]^T$$



# Filtrage sur graphe

**Définition** (*filtre sur graphe*) :

Les *filtres sur graphe* (invariants par rapport aux sommets), sont des opérateurs linéaires sur les signaux sur graphes  $\mathbf{H} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , de la forme :

$$\mathbf{H} \triangleq \sum_{k=0}^{m-1} c_k \mathbf{L}^k$$

avec

$m - 1$  : ordre du filtre

$\mathbf{c} = [c_0, c_1, \dots, c_{m-1}]^T \in \mathbb{R}^m$  : vecteur des coefficients du filtre

**Remarque :**

On peut observer que les filtres sur graphes sont des *polynômes* de la matrice laplacienne  $\mathbf{L}$  de degré  $m - 1$

Exemple (Filtre d'ordre 3) :

$$\mathbf{H} = 2\mathbf{I}_n + 3\mathbf{L} - 4\mathbf{L}^2 + 7\mathbf{L}^3$$

avec

$$\mathbf{c} = [c_0, c_1, c_2, c_3]^T = [2, 3, -4, 7]^T$$



# Filtrage sur graphe

- On peut réécrire le filtre sur graphe  $\mathbf{H}$  de façon équivalente comme suit :

$$\mathbf{H} = \mathbf{U} \left( \sum_{k=0}^{m-1} c_k \Lambda^k \right) \mathbf{U}^T$$

- Nous pouvons interpréter la matrice diagonale

$$\hat{\mathbf{H}} \triangleq \sum_{k=0}^{m-1} c_k \Lambda^k$$

comme la *réponse en fréquence* de  $\mathbf{H}$ . On peut réécrire cette matrice comme  $\hat{\mathbf{H}} = \text{diag}(\hat{\mathbf{c}})$  où le vecteur  $\hat{\mathbf{c}} \in \mathbb{R}^n$  contient les  $n$  réponses en fréquence du filtre sur graphe

- Le **filtrage d'un signal**  $\mathbf{f}$  par le filtre  $\mathbf{H}$ , s'écrit donc :

$$\mathbf{f}^h = \mathbf{U} \hat{\mathbf{H}} \mathbf{U}^T \mathbf{f}$$

- “*Optimal graph-filter design and applications to distributed linear network operators*”, S. Segarra, A.G. Marques, A. Ribeiro, IEEE Trans. Signal Process., vol. 65, n. 15, pp. 4117-4131, 2017
- “*Design of graph filters and filterbanks*”, N. Tremblay, P. Gonçalves, P. Borgnat. Ch. 11, in Cooperative and Graph Signal Processing, pp. 299-324, Academic Press, 2018



# Echantillonnage sur graphe

- Une *théorie de l'échantillonnage sur graphe* est de grand intérêt pour de nombreuses applications pratiques

## Exemple :

Étant donné un graphe qui représente les connexions entre des amis sur **Facebook**, on peut décider d'échantillonner une fraction des utilisateurs, trouver leurs hobbies et ensuite essayer de reconstruire les hobbies de *tous* les utilisateurs



## Définition (*Sous-échantillonnage d'un signal sur graphe*)

L'*opération de sous-échantillonnage* d'un signal sur graphe  $f$  est définie comme la restriction du signal  $f$  sur un certain sous-ensemble des sommets  $S \subset V$  de  $\mathcal{G}$ , appelé *ensemble de sous-échantillonnage*.

Le signal sous-échantillonné est noté  $f(S)$  : il est un vecteur de taille réduite  $m < n$  où  $n$  est le nombre de sommets de  $\mathcal{G}$



# Echantillonnage sur graphe

## Problème *direct* :

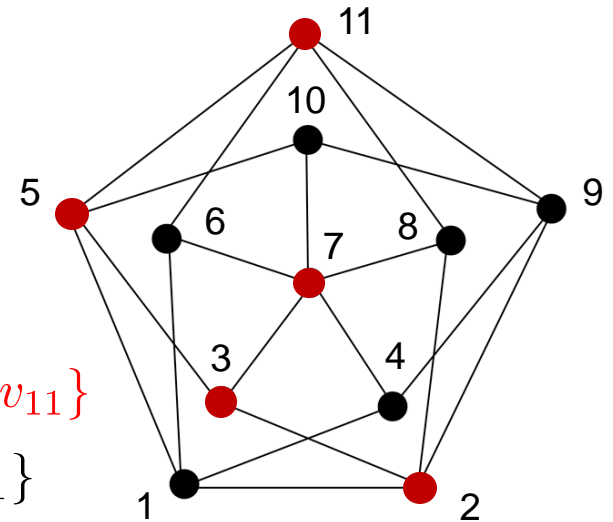
Déterminer les ensembles de sous-échantillonnage qui permettent de reconstruire de *façon unique* un signal  $f$  avec une bande passante  $\omega_c$  donnée (cf. théorème d'échantillonnage ou théorème de Shannon en Théorie du Signal classique)

## Problème *inverse* :

Étant donné un signal  $f$  avec une bande passante  $\omega_c$  déterminer quel est *le plus petit sous-ensemble* des sommets du graphe  $S_{\text{opt}}$  tel que le signal est représenté *de façon unique* par  $f(S_{\text{opt}})$

$$S_{\text{opt}} = \{v_2, v_3, v_5, v_7, v_{11}\}$$

$$V = \{v_1, v_2, \dots, v_{11}\}$$



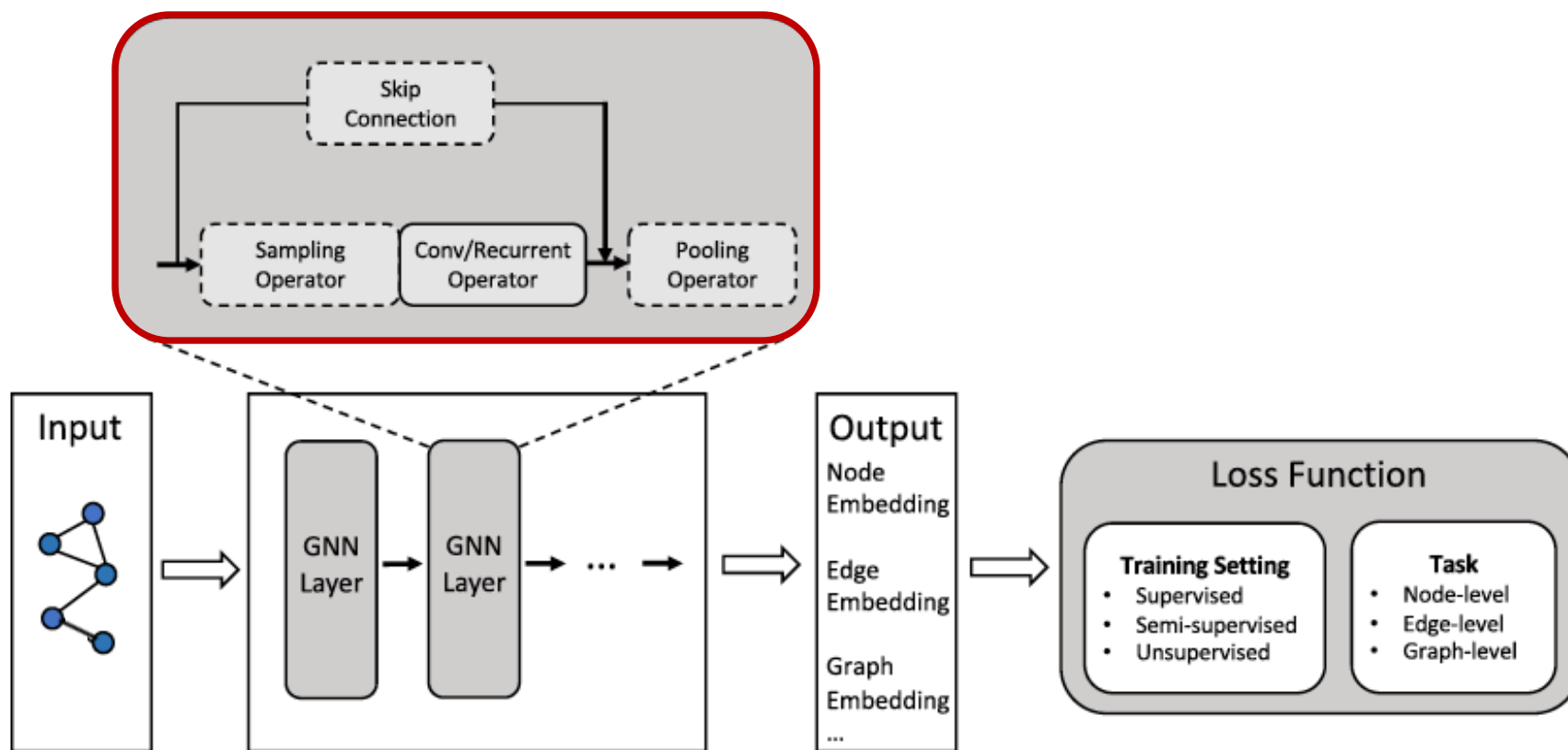
- Théorie de l'échantillonnage pour signaux définis uniquement sur *graphes non orientés* :  
“Towards a sampling theorem for signals on arbitrary graphs”, A. Anis, A. Gadde, A. Ortega, in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp. 3864-3868, 2014
- Théorie de l'échantillonnage pour signaux définis sur *graphes non orientés* et *orientés* :  
“Discrete Signal Processing on Graphs: Sampling Theory”, S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, IEEE Trans. Signal Processing, vol. 63, n. 24, pp. 6510-6523, 2015





# Réseaux de neurones sur graphes (GNN)

- Structure générale des réseaux de neurones sur graphes (GNN)

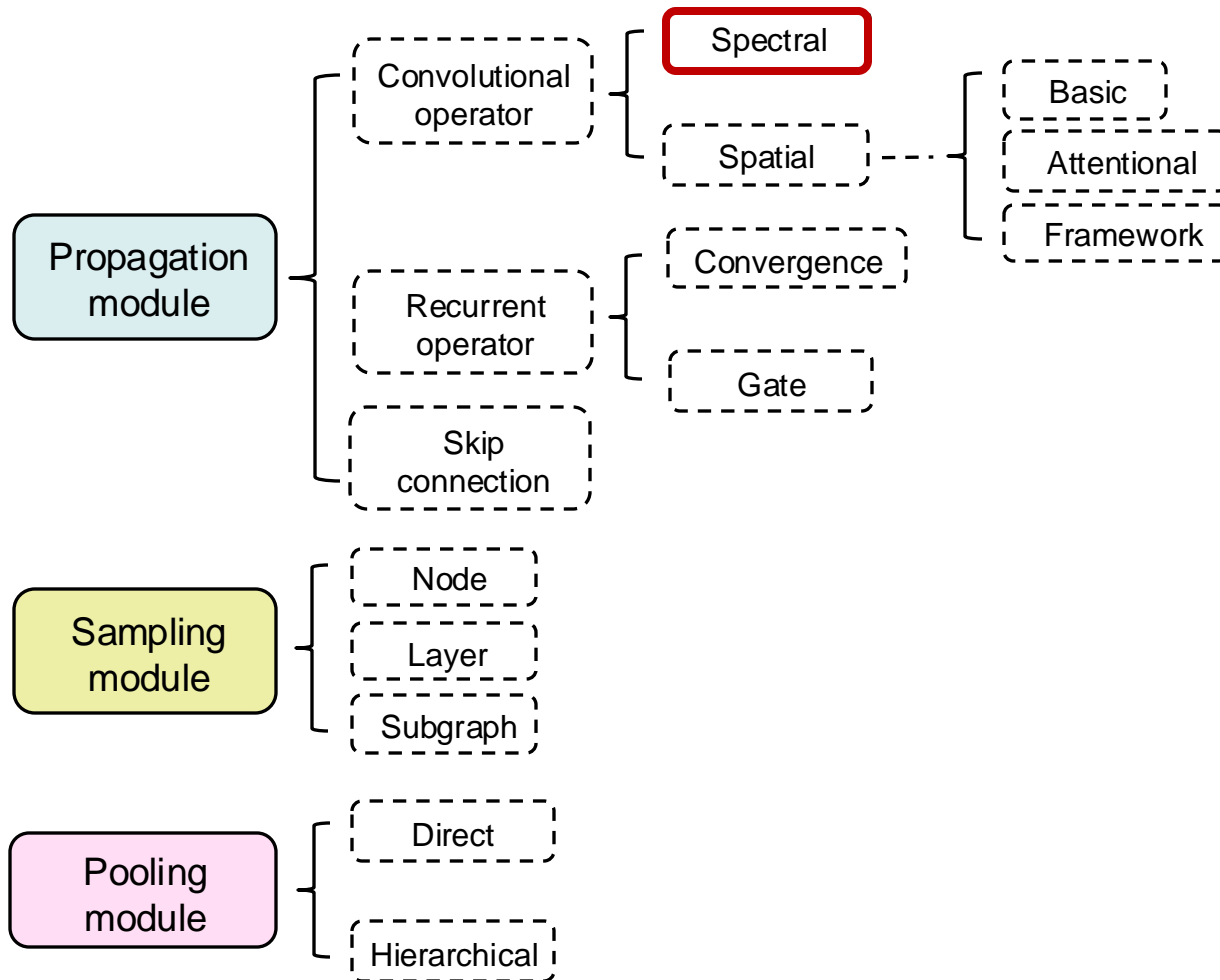


- “Graph neural networks: A review of methods and applications“, J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, AI Open, vol, 1, pp. 57-81, 2020



# Réseaux de neurones sur graphes (GNN)

- Vue d'ensemble des modules computationnels d'un GNN



# Réseaux de neurones sur graphes (GNN)

- Nous avons vu que la *convolution sur graphe* de deux signaux  $f(x)$  et  $g(x)$  s'écrit :

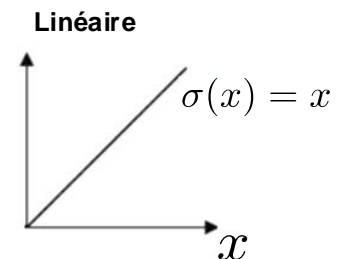
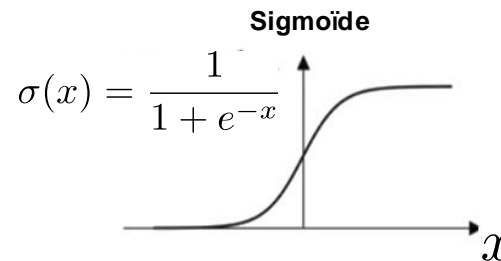
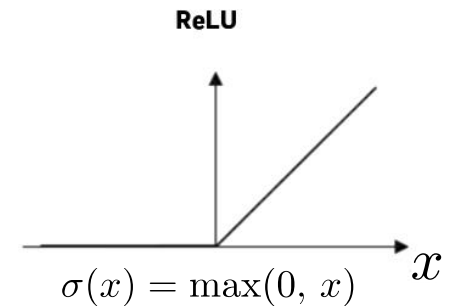
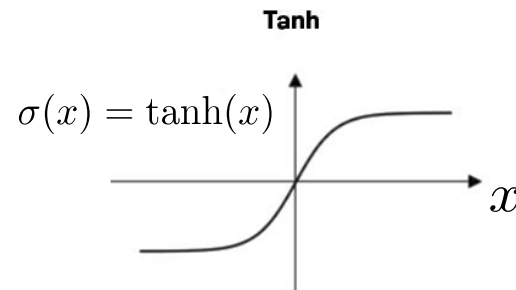
$$\mathbf{f} *_G \mathbf{g} \triangleq \mathcal{F}^{-1} [\mathcal{F}[\mathbf{f}] \circ \mathcal{F}[\mathbf{g}]] = \mathbf{U}(\mathbf{U}^T \mathbf{f} \circ \mathbf{U}^T \mathbf{g})$$

- Nous pouvons simplifier l'expression  $\mathbf{U}^T \mathbf{f}$ , en utilisant une matrice diagonale  $\mathbf{W}$  (« filtre ») calculable par **apprentissage**. Nous avons donc la formule suivante des *méthodes spectrales*, pour le calcul de la convolution :

$$\mathbf{f}_w *_G \mathbf{g} = \mathbf{U} \mathbf{W} \mathbf{U}^T \mathbf{g}$$

## Remarque :

Dans l'expression précédente, on retrouve souvent aussi une *fonction non linéaire d'activation*  $\sigma(\cdot)$  appliquée aux valeurs du signal sur graphe (par ex. une sigmoïde ou une fonction ReLU, « Rectified Linear Unit »)



# Réseaux de neurones sur graphes (GNN)

- Plusieurs façons de définir le filtre  $\mathbf{W}$  :
  - Spectral Network [Bruna *et al.*, 2014] ( $\mathbf{W} = \text{diag}(\mathbf{w})$  obtenu par apprentissage)
  - ChebNet [Defferrard *et al.*, 2016] (filtres polynomiaux de Chebyshev)
  - CayleyNets [Levie *et al.*, 2019] (filtres polynomiaux de Cayley)
  - ARMA [Bianchi *et al.*, 2022] (filtres rationnels ARMA, “AutoRegressive Moving Average”)
  - GCN (Graph Convolutional Network) [Kipf & Welling, 2017]
  - AGCN (Adaptive Graph Convolution Network) [Li *et al.*, 2018]
  - DGCN (Dual Graph Convolutional Network) [Zhuang & Ma, 2018]
  - GWNN (Graph Wavelet Neural Network) [Xu *et al.*, 2019]

AGCN et DGCN essaient d’améliorer les méthodes spectrales en « augmentant » la matrice laplacienne, tandis que GWNN fournit une *alternative* à la transformée de Fourier

## Articles *survey* sur les GNN :

- “*Graph neural networks: A review of methods and applications*“, J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, AI Open, vol. 1, pp. 57-81, 2020
- “*Graphs, Convolutions, and Neural Networks: From graph filters to graph neural networks*“ F. Gama, E. Isufi, G. Leus, A. Ribeiro, IEEE Signal Process. Mag., vol. 37, n. 6, pp. 128-138, 2020
- “*Graph signal processing for machine learning: A review and new perspectives*“, X. Dong, D. Thanou, L. Toni, M. Bronstein, P. Frossard, IEEE Signal Process. Mag., vol. 37, n. 6, pp. 117-127, 2020
- “*A Comprehensive Survey on Graph Neural Networks*“, Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, IEEE Trans. Neural Networks Learn. Syst., vol. 32, n. 1, pp. 4-24, 2021



# D'autres axes de recherche et outils logiciels

- **Compression et reconstruction de signaux sur graphe**
  - “*Graph-Based Compression of Dynamic 3D Point Cloud Sequences*“, D. Thanou, P.A. Chou, P. Frossard, IEEE Trans. Image Processing, vol. 25, n. 4, pp. 1765-1778, 2016
  - “*Graph-signal Reconstruction and Blind Deconvolution for Structured Inputs*“, D. Ramírez, A.G. Marques, S. Segarra, Signal Processing, vol. 188, p. 108180, 2021
- **Transformée en ondelettes sur graphe (“graph wavelet transform”)**
  - “*Wavelets on graphs via spectral graph theory*“, D. K. Hammond, P. Vandergheynst, R. Gribonval, Appl. Comput. Harmon. Anal., vol. 30, n. 2, pp. 129-150, 2011
  - “*Design of graph filters and filterbanks*“, N. Tremblay, P. Gonçalves, P. Borgnat. Ch. 11, in Cooperative and Graph Signal Processing, pp. 299-324, Academic Press, 2018
- **Toolbox Matlab pour le traitement du signal sur graphe**
  - “*GSPBOX: A toolbox for signal processing on graphs*“, N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, D.K. Hammond, Arxiv e-print, 08-2014, <https://epfl-lts2.github.io/gspbox-html>
  - “*GraSP: A Matlab Toolbox for Graph Signal Processing*“, B. Girault, S.S. Narayanan, A. Ortega, P. Gonçalves, E. Fleury, in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp. 6574-6575, 2017, <https://gitlab.inria.fr/bgirault/grasp>

