Topological Deep Learning

Part 2: Sheaf Neural Networks

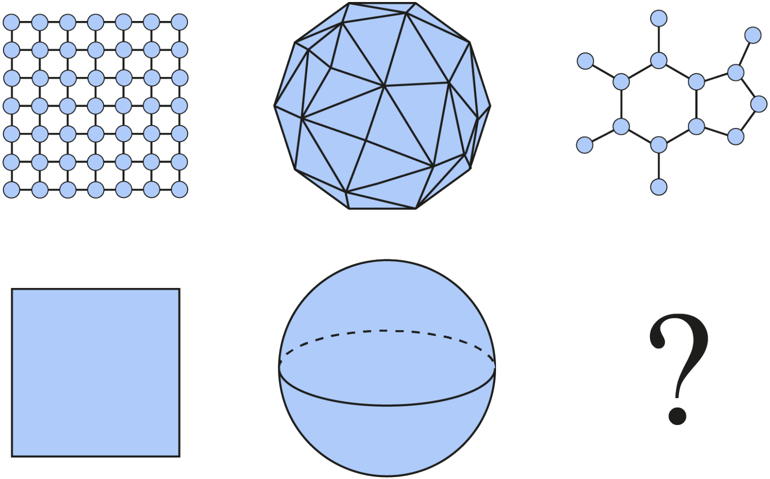
Cristian Bodnar
University of Cambridge

Geometric Deep Learning Summer School
Pescara, Italy
July 26, 2022

The Big Picture

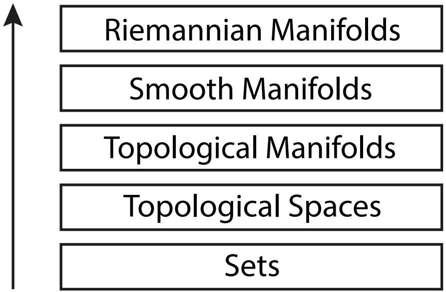
Geometry on graphs

Graphs, the most prevalent space in GDL do not have a "natural" geometric structure[1]



[1] The diagram is inspired from Bronstein, *Graph Neural Networks through the lens of Differential Geometry and Algebraic Topology*, 2022

A bottom up approach

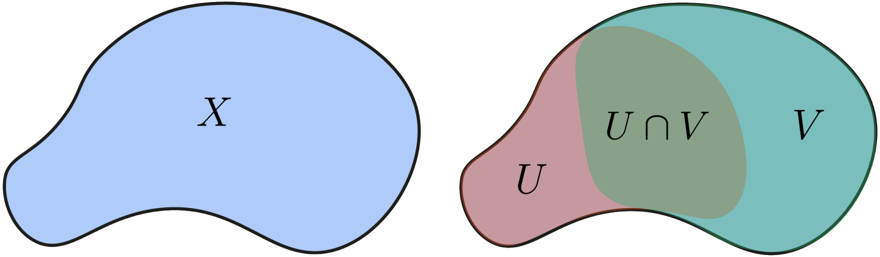


In this talk, starting from topological spaces, we will work our way up to recover some of the elements of this hierarchy.

Topological Spaces

A *topological space* is a set X together with a collection $\mathcal{T}$ of subsets of X called the *open sets* of X and satisfying certain axioms:

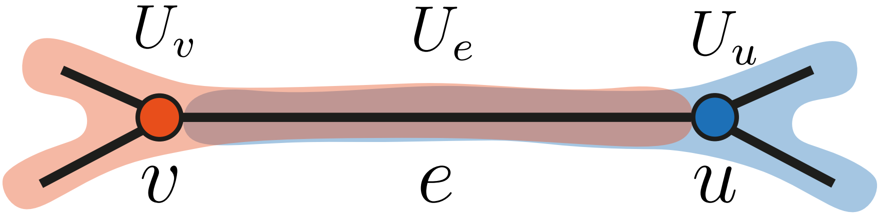
1. The empty set and X belong to $\mathcal{T}$.
2. Any finite intersection and arbitrary union of open sets is an open set.



A topological space X and its open sets. These sets provide neighbourhood structure for the points of X.

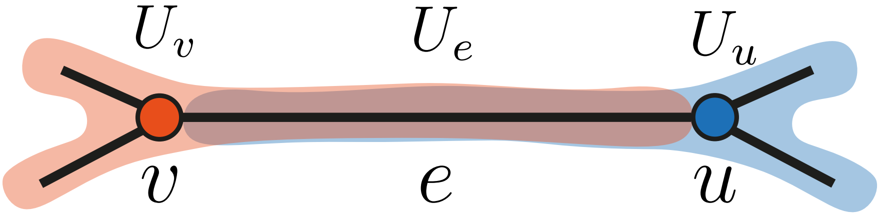
Graphs as Topological Spaces

We can see graphs as topological spaces. Each node v (and edge e) has an associated open set U_v (and U_e, respectively). These are given by the star operator we saw in the previous talk.



Graphs as Topological Spaces

We can see graphs as topological spaces. Each node v (and edge e) has an associated open set U_v (and U_e, respectively). These are given by the star operator we saw in the previous talk.

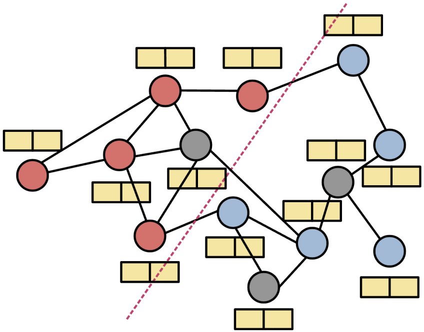


In what follows, we will see how this perspective can help us understand and develop better GNNs.

Heterophily and Oversmoothing in GNNs

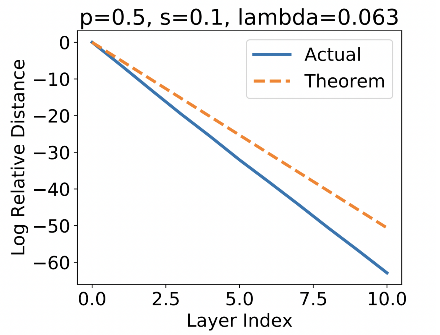
Node classification

Let $G = (V, E)$ be a graph with n nodes and a node feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. We want to label the gray nodes by training on the red and blue nodes.



The Oversmoothing Problem

In some GNNs[2] features become progressively smoother with increased depth.

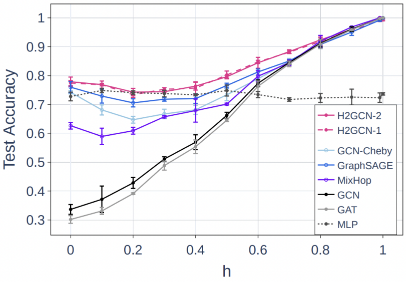


With more layers, GCN approaches a "smooth" subspace where all the node features are constant[1].

[2] Oono and Suzuki, "Graph neural networks exponentially lose expressive power for node classification", 2019; Cai and Wang, "A note on over-smoothing for graph neural networks", 2020.

The Heterophily Problem

It was remarked[3] that GNNs struggle in heterophilic settings (i.e. graphs where a node tends to be connected to nodes belonging to other classes).



The performance of GNNs is strongly correlated to the homophily level of a graph[2].

[3] Zhu et al., "Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs", 2020.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

Heat diffusion is a simple process suffering from oversmoothing and heterophily:

$$\dot{\mathbf{X}}(t) = -\Delta_0 \mathbf{X}(t)$$

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

Heat diffusion is a simple process suffering from oversmoothing and heterophily:

$$\dot{\mathbf{X}}(t) = -\Delta_0 \mathbf{X}(t) \rightsquigarrow \mathbf{X}(t+1) = \mathbf{X}(t) - \Delta_0 \mathbf{X}(t)$$

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

Heat diffusion is a simple process suffering from oversmoothing and heterophily:

$$\dot{\mathbf{X}}(t) = -\Delta_0 \mathbf{X}(t) \rightsquigarrow \mathbf{X}(t+1) = \mathbf{X}(t) - \Delta_0 \mathbf{X}(t) = (\mathbf{I} - \Delta_0)\mathbf{X}(t) \tag{1}$$

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

Heat diffusion is a simple process suffering from oversmoothing and heterophily:

$$\dot{\mathbf{X}}(t) = -\Delta_0 \mathbf{X}(t) \rightsquigarrow \mathbf{X}(t+1) = \mathbf{X}(t) - \Delta_0 \mathbf{X}(t) = (\mathbf{I} - \Delta_0)\mathbf{X}(t) \tag{1}$$

This is strikingly similar to GCN[4]:

$$\mathrm{GCN}(\mathbf{X}, \mathbf{A}) := \sigma\left(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{X}\mathbf{W}\right)$$

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

Heat diffusion is a simple process suffering from oversmoothing and heterophily:

$$\dot{\mathbf{X}}(t) = -\Delta_0 \mathbf{X}(t) \rightsquigarrow \mathbf{X}(t+1) = \mathbf{X}(t) - \Delta_0 \mathbf{X}(t) = (\mathbf{I} - \Delta_0)\mathbf{X}(t) \tag{1}$$

This is strikingly similar to GCN[4]:

$$\mathrm{GCN}(\mathbf{X}, \mathbf{A}) := \sigma\left(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{X}\mathbf{W}\right) = \sigma\left((\mathbf{I} - \Delta_0)\mathbf{X}\mathbf{W}\right) \tag{2}$$

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

From Heat Diffusion to Graph Convolutions

Let G be a graph with self loops, degree matrix $\mathbf{D}$, adjacency matrix $\mathbf{A}$, normalised Laplacian $\Delta_0 := I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, and node features $\mathbf{X} \in \mathbb{R}^{n \times d}$.

Heat diffusion is a simple process suffering from oversmoothing and heterophily:

$$\dot{\mathbf{X}}(t) = -\Delta_0 \mathbf{X}(t) \rightsquigarrow \mathbf{X}(t+1) = \mathbf{X}(t) - \Delta_0 \mathbf{X}(t) = (\mathbf{I} - \Delta_0)\mathbf{X}(t) \tag{1}$$

This is strikingly similar to GCN[4]:

$$\mathrm{GCN}(\mathbf{X}, \mathbf{A}) := \sigma\left(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{X}\mathbf{W}\right) = \sigma\left((\mathbf{I} - \Delta_0)\mathbf{X}\mathbf{W}\right) \tag{2}$$

Question

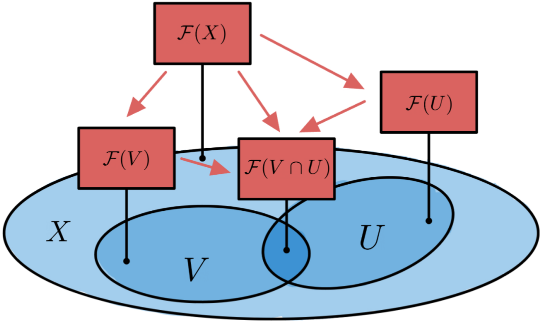
How can we make the base diffusion process more powerful?

[4] Kipf and Welling, "Semi-supervised classification with graph convolutional networks", 2017.

Sheaves

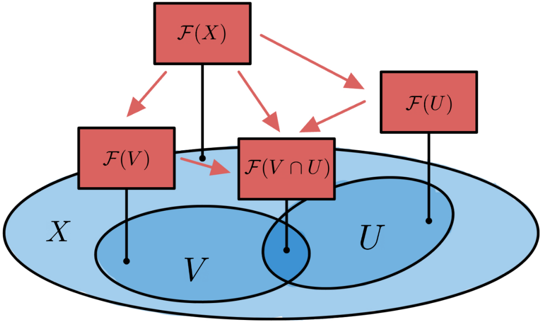
Presheaves

Informally, *Pre-sheaves* are an assignment of some data to the open sets of a space X.



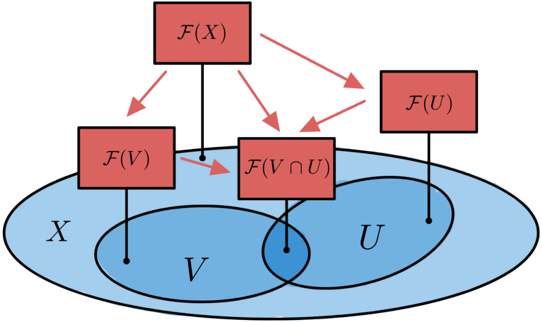
Presheaves

Informally, *Pre-sheaves* are an assignment of some data to the open sets of a space X.

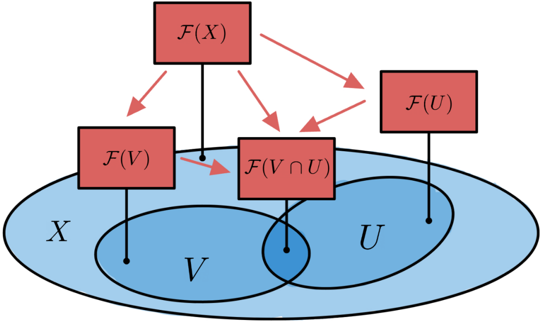


For each open set U, we denote the data attached to it by $\mathcal{F}(U)$. Whenever $U \subseteq X$, we can follow an arrow $\mathcal{F}(X) \to \mathcal{F}(U)$ to "restrict" the data of X to a smaller region U.

Presheaves



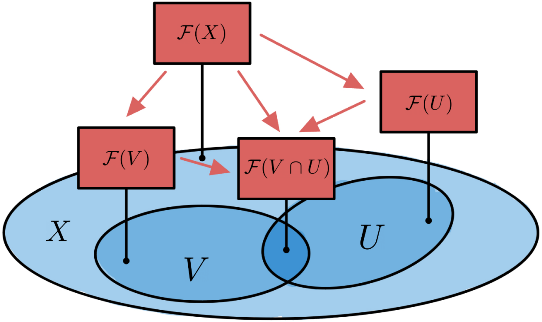
Presheaves



A *presheaf* of sets $\mathcal{F}$ over a topological space X consists of:

1. For each open set $U \subseteq X$, a set $\mathcal{F}(U)$. Its elements are called *sections* of $\mathcal{F}$ over U.

Presheaves



A *presheaf* of sets $\mathcal{F}$ over a topological space X consists of:

1. For each open set $U \subseteq X$, a set $\mathcal{F}(U)$. Its elements are called *sections* of $\mathcal{F}$ over U.
2. For each inclusion of open sets $V \subseteq U$, a function $\mathcal{F}_{U,V} : \mathcal{F}(U) \to \mathcal{F}(V)$ with the property that if $W \subseteq V \subseteq U$, then $\mathcal{F}_{U,W} = \mathcal{F}_{V,W} \circ \mathcal{F}_{U,V}$ and $\mathcal{F}_{U,U} = \mathrm{id}_{\mathcal{F}(\mathrm{U})}$.

The presheaf of continuous functions over $\mathbb{R}$

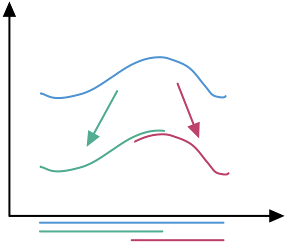
Let $X = \mathbb{R}$, $\mathcal{F}(U) = \{f : U \to \mathbb{R} \mid f \text{ is continous}\}$ and let $V \subseteq U$ $\mathcal{F}_{U:V} : \mathcal{F}(U) \to \mathcal{F}(V)$ be the restriction map sending $f \mapsto f|_V$. This presheaf satisfies two nice properties:

The presheaf of continuous functions over $\mathbb{R}$

Let $X = \mathbb{R}$, $\mathcal{F}(U) = \{f : U \to \mathbb{R} \mid f \text{ is continous}\}$ and let $V \subseteq U$ $\mathcal{F}_{U:V} : \mathcal{F}(U) \to \mathcal{F}(V)$ be the restriction map sending $f \mapsto f|_V$. This presheaf satisfies two nice properties:

Locality

Let $(U_i)_{i \in I}$ be an open cover for U. For any $s, t \in \mathcal{F}(U)$, if $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.

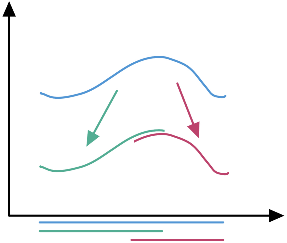


The presheaf of continuous functions over $\mathbb{R}$

Let $X = \mathbb{R}$, $\mathcal{F}(U) = \{f : U \to \mathbb{R} \mid f \text{ is continous}\}$ and let $V \subseteq U$ $\mathcal{F}_{U:V} : \mathcal{F}(U) \to \mathcal{F}(V)$ be the restriction map sending $f \mapsto f|_V$. This presheaf satisfies two nice properties:

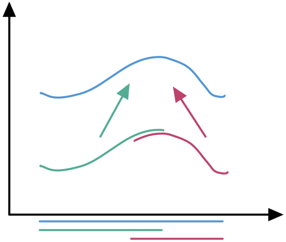
Locality

Let $(U_i)_{i \in I}$ be an open cover for U. For any $s, t \in \mathcal{F}(U)$, if $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.



Glueing

Let $(U_i)_{i \in I}$ be an open cover for U and $(s_i \in \mathcal{F}(U_i))_{i \in I}$. If for all $i, j \in I$, $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$, then there is a unique $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$.



Sheaves

For $s \in \mathcal{F}(U)$ and $V \subseteq U$, denote by $s|_V := \mathcal{F}_{U,V}(s)$.

Sheaves

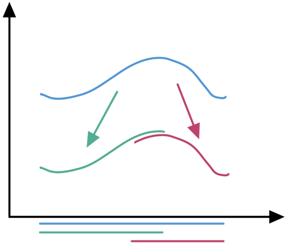
For $s \in \mathcal{F}(U)$ and $V \subseteq U$, denote by $s|_V := \mathcal{F}_{U,V}(s)$. Then, a *sheaf* is a presheaf that also satisfies the locality and glueing conditions.

Sheaves

For $s \in \mathcal{F}(U)$ and $V \subseteq U$, denote by $s|_V := \mathcal{F}_{U,V}(s)$. Then, a *sheaf* is a presheaf that also satisfies the locality and glueing conditions.

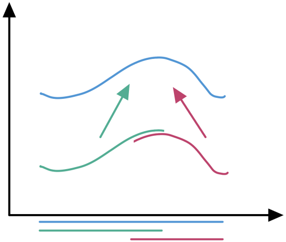
Locality

Let $(U_i)_{i\in I}$ be an open cover for U. For any $s, t \in \mathcal{F}(U)$, if $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.



Glueing

Let $(U_i)_{i\in I}$ be an open cover for U and $(s_i \in \mathcal{F}(U_i))_{i\in I}$. If for all $i, j \in I$, $s_i|_{U_i\cap U_j} = s_j|_{U_i\cap U_j}$, then there is a unique $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$.

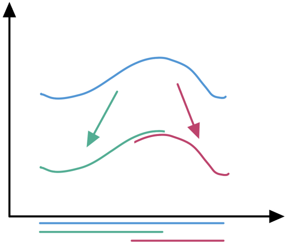


Sheaves

For $s \in \mathcal{F}(U)$ and $V \subseteq U$, denote by $s|_V := \mathcal{F}_{U,V}(s)$. Then, a *sheaf* is a presheaf that also satisfies the locality and glueing conditions.

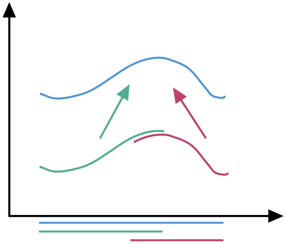
Locality

Let $(U_i)_{i\in I}$ be an open cover for U. For any $s, t \in \mathcal{F}(U)$, if $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.

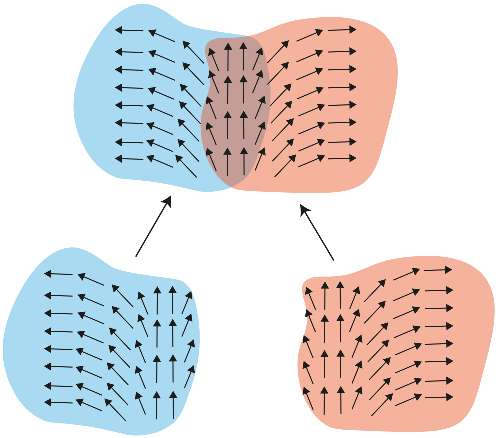


Glueing

Let $(U_i)_{i\in I}$ be an open cover for U and $(s_i \in \mathcal{F}(U_i))_{i\in I}$. If for all $i, j \in I$, $s_i|_{U_i\cap U_j} = s_j|_{U_i\cap U_j}$, then there is a unique $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$.



Examples of sheaves



We can glue two vector fields that agree on their overlap.

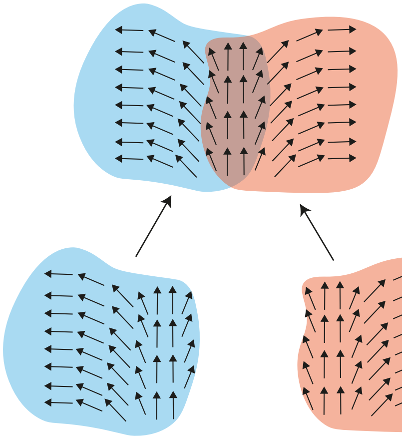
We have already seen that the presheaf of continuous functions over $\mathbb{R}$ is a sheaf.

Another example is the sheaf of vector fields over a smooth manifold M.

$$\mathcal{F}(U) = \{f : U \to TU \mid f \text{ is a vector field}\}$$

The restriction maps are simply restrictions of the vector field.

Examples of sheaves



We can glue two vector fields that agree on their overlap.

We have already seen that the presheaf of continuous functions over $\mathbb{R}$ is a sheaf.

Another example is the sheaf of vector fields over a smooth manifold M.

$$\mathcal{F}(U) = \{f : U \to TU \mid f \text{ is a vector field}\}$$

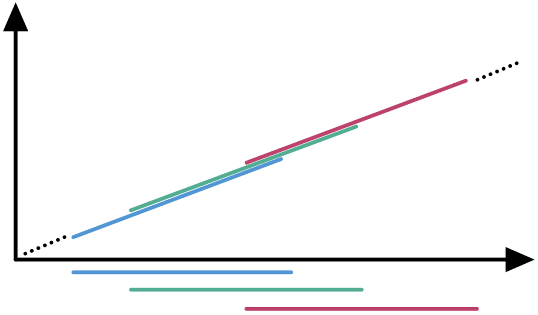
The restriction maps are simply restrictions of the vector field.

Key idea

Sheaves are about creating big(ger) data from small(er) data.

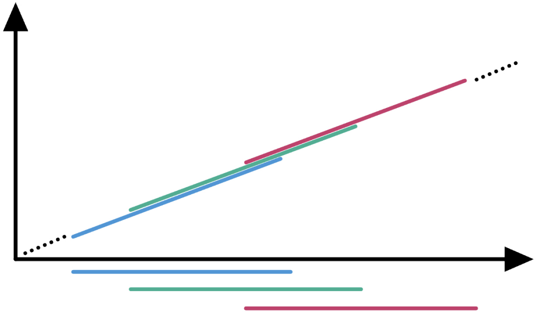
Presheaves that are not sheaves

Let $X = \mathbb{R}$ and consider the assignment $\mathcal{F}(U) = \{f : U \to \mathbb{R} \mid f \text{ is cont. and bounded}\}$. Then if we glue over $\mathbb{R}$ infinitely many functions bounded on subsets of $\mathbb{R}$, we can easily obtain an ubounded function over $\mathbb{R}$. Therefore, this is not a sheaf.



Presheaves that are not sheaves

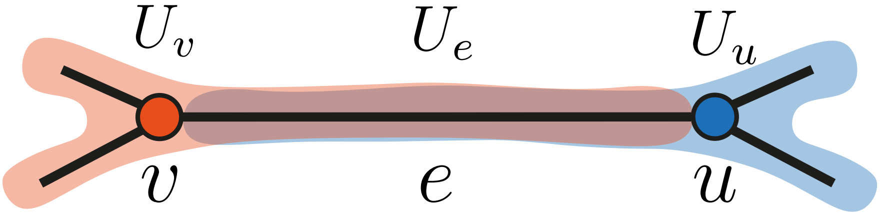
Let $X = \mathbb{R}$ and consider the assignment $\mathcal{F}(U) = \{f : U \to \mathbb{R} \mid f \text{ is cont. and bounded}\}$. Then if we glue over $\mathbb{R}$ infinitely many functions bounded on subsets of $\mathbb{R}$, we can easily obtain an ubounded function over $\mathbb{R}$. Therefore, this is not a sheaf.



Another example is the presheaf of constant functions over $\mathbb{R}$:
$\mathcal{F}(U) = \{f : U \to \mathbb{R} \mid f \text{ is constant}\}$. Can you see why?

Towards Sheaves on Graphs

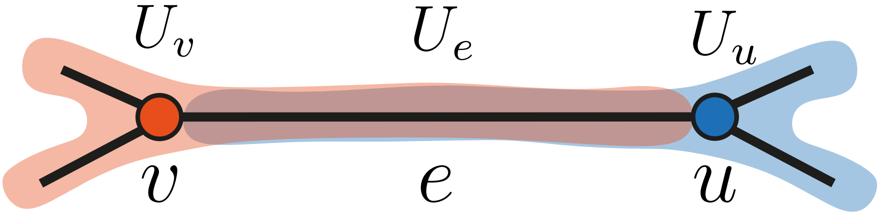
Given a topological space X, an *open base* is a collection of open subsets $\mathcal{B}$ such that any other open subsets of X can be expressed as a union of subsets in $\mathcal{B}$. The elements of $\mathcal{B}$ are called *basic open sets*.



[5] Hu, "A Brief Note for Sheaf Structures on Posets", 2020.

Towards Sheaves on Graphs

Given a topological space X, an *open base* is a collection of open subsets $\mathcal{B}$ such that any other open subsets of X can be expressed as a union of subsets in $\mathcal{B}$. The elements of $\mathcal{B}$ are called *basic open sets*.



We will now consider graphs with the topology generated by the basis of open stars of all the vertices and edges. We will show how this space can be equipped with a sheaf, following an approach inspired by CS Hu[5].

[5] Hu, "A Brief Note for Sheaf Structures on Posets", 2020.

Base presheaves on Graphs

A $\mathcal{B}$-*presheaf* is a presheaf where we only care about basic open sets. More formally:

1. For each open set $U \in \mathcal{B}$, a set $\mathcal{F}(U)$.

Base presheaves on Graphs

A $\mathcal{B}$-*presheaf* is a presheaf where we only care about basic open sets. More formally:

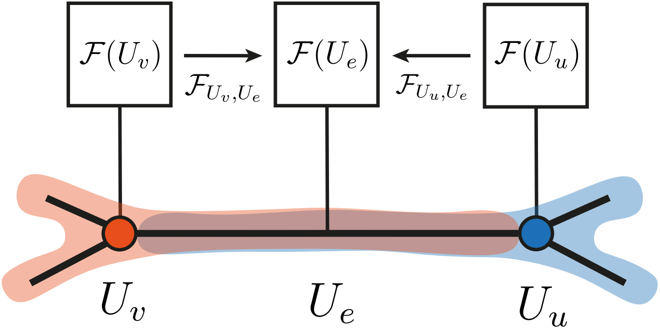
1. For each open set $U \in \mathcal{B}$, a set $\mathcal{F}(U)$.
2. For each pair $V \subseteq U$ of members of $\mathcal{B}$, a function $\mathcal{F}_{U,V} : \mathcal{F}(U) \to \mathcal{F}(V)$ with the usual properties.

Base presheaves on Graphs

A $\mathcal{B}$-*presheaf* is a presheaf where we only care about basic open sets. More formally:

1. For each open set $U \in \mathcal{B}$, a set $\mathcal{F}(U)$.
2. For each pair $V \subseteq U$ of members of $\mathcal{B}$, a function $\mathcal{F}_{U,V} : \mathcal{F}(U) \to \mathcal{F}(V)$ with the usual properties.

Now, we can easily define a $\mathcal{B}$-*presheaf* on graphs:



Base sheaves on Graphs

We can similarly define a $\mathcal{B}$*-sheaf*. It is a $\mathcal{B}$*-presheaf* such that it satisfies:

1. **Locality**: Let $U \in \mathcal{B}$ and $s, t \in \mathcal{F}(U)$. If U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ such that $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.

Base sheaves on Graphs

We can similarly define a $\mathcal{B}$-*sheaf*. It is a $\mathcal{B}$-*presheaf* such that it satisfies:

1. **Locality**: Let $U \in \mathcal{B}$ and $s, t \in \mathcal{F}(U)$. If U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ such that $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.
2. **Glueing**: Suppose $U \in \mathcal{B}$ and U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ with local sections $s_i \in \mathcal{F}(U_i)$ such that for all $i, j \in I$, $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$. Then there is an $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$

Base sheaves on Graphs

We can similarly define a $\mathcal{B}$*-sheaf*. It is a $\mathcal{B}$*-presheaf* such that it satisfies:

1. **Locality**: Let $U \in \mathcal{B}$ and $s, t \in \mathcal{F}(U)$. If U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ such that $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.
2. **Glueing**: Suppose $U \in \mathcal{B}$ and U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ with local sections $s_i \in \mathcal{F}(U_i)$ such that for all $i, j \in I$, $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$. Then there is an $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$

Now a little miracle happens:

Theorem

Any $\mathcal{B}$-presheaf on a graph with the topology generated by the open stars is a $\mathcal{B}$-sheaf.

Proving Locality

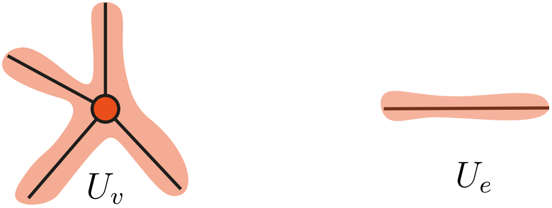
Locality: Let $U \in \mathcal{B}$ and $s, t \in \mathcal{F}(U)$. If U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ such that $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.

Proving Locality

Locality: Let $U \in \mathcal{B}$ and $s, t \in \mathcal{F}(U)$. If U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ such that $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.

Proof.

Notice that we only have only two types of open sets in $\mathcal{B}$:

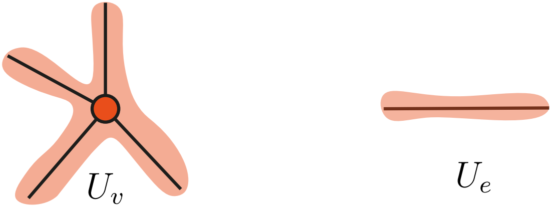


For U_e the only cover is the trivial one: $\{U_e\}$. Thus, $s = s|_{U_e} = t|_{U_e} = t$.

Proving Locality

Locality: Let $U \in \mathcal{B}$ and $s, t \in \mathcal{F}(U)$. If U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ such that $s|_{U_i} = t|_{U_i}$ for all $i \in I$, then $s = t$.

Proof.
Notice that we only have only two types of open sets in $\mathcal{B}$:



For U_e the only cover is the trivial one: $\{U_e\}$. Thus, $s = s|_{U_e} = t|_{U_e} = t$. For U_v, notice that $U_v \in (U_i)_{i \in I}$ because the vertex v cannot be covered by other open sets in $\mathcal{B}$. Then again we have $s = s|_{U_v} = t|_{U_v} = t$. $\square$

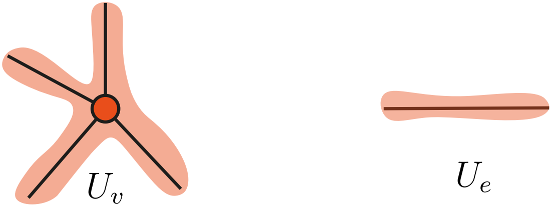
Proving Glueing

Glueing: Suppose $U \in \mathcal{B}$ and U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ with local sections $s_i \in \mathcal{F}(U_i)$ such that for all $i, j \in I$, $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$. Then there is an $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$

Proving Glueing

Glueing: Suppose $U \in \mathcal{B}$ and U is covered by $(U_i)_{i \in I} \subseteq \mathcal{B}$ with local sections $s_i \in \mathcal{F}(U_i)$ such that for all $i, j \in I$, $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$. Then there is an $s \in \mathcal{F}(U)$ such that $s|_{U_i} = s_i$ for all $i \in I$

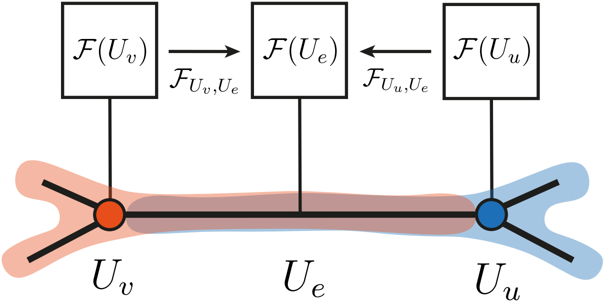
Proof.



As before, for open sets of type U_e, the proof is trivial. For open sets of type U_v we exploit again that $U_v = U_k$ for some $k \in I$. Let $s_k \in \mathcal{F}(U_k) = \mathcal{F}(U_v)$. We have that $s_k|_{U_i} = s_k|_{U_k \cap U_i} = s_i|_{U_k \cap U_i} = s_i|_{U_v \cap U_i} = s_i|_{U_i} = s_i$ $\square$

Sheaves on Graphs

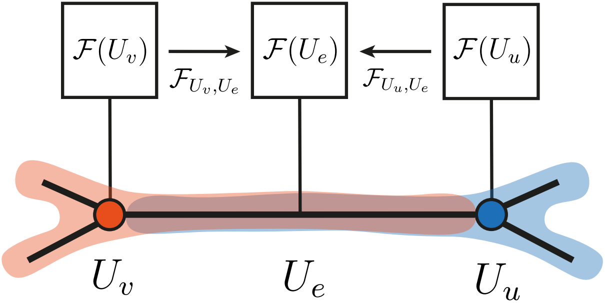
Now we know that our $\mathcal{B}$*-presheaf* is a $\mathcal{B}$-sheaf. But how do we get a sheaf now?



To each open star we assign a set and for each $U_e \subseteq U_v$ a restriction map.

Sheaves on Graphs

Now we know that our $\mathcal{B}$*-presheaf* is a $\mathcal{B}$-sheaf. But how do we get a sheaf now?



To each open star we assign a set and for each $U_e \subseteq U_v$ a restriction map.

It turns out, that sheaves behave a bit like a linear operator. It is sufficient to specify how it behaves on a basis to fully specify its behaviour.

Sheaves on Graphs

Theorem

A $\mathcal{B}$-sheaf $\mathcal{F}$ on a space X uniquely induces a sheaf $\mathcal{F}^+$ on X such that $\mathcal{F}$ and $\mathcal{F}^+$ are canonically isomorphic.

Sheaves on Graphs

Theorem

A $\mathcal{B}$-sheaf $\mathcal{F}$ on a space X uniquely induces a sheaf $\mathcal{F}^+$ on X such that $\mathcal{F}$ and $\mathcal{F}^+$ are canonically isomorphic.

Let $I(U) = \{V \mid V \in \mathcal{B}, V \subseteq U\}$. Then the sections of $\mathcal{F}^+$ are constructed as follows:

$$\mathcal{F}^+(U) := \left\{(s_V)_{V\in I(U)} \in \prod_{V\in I(U)} \mathcal{F}(V) \mid \mathcal{F}_{V,W}(s_V) = s_W,\ \forall W \subseteq V\right\}$$

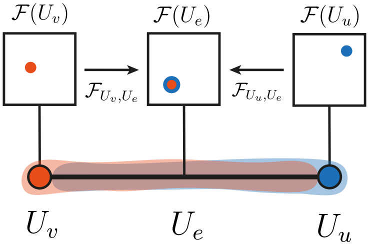
Sheaves on Graphs

Theorem

A $\mathcal{B}$-sheaf $\mathcal{F}$ on a space X uniquely induces a sheaf $\mathcal{F}^+$ on X such that $\mathcal{F}$ and $\mathcal{F}^+$ are canonically isomorphic.

Let $I(U) = \{V \mid V \in \mathcal{B}, V \subseteq U\}$. Then the sections of $\mathcal{F}^+$ are constructed as follows:

$$\mathcal{F}^+(U) := \left\{ (s_V)_{V \in I(U)} \in \prod_{V \in I(U)} \mathcal{F}(V) \mid \mathcal{F}_{V,W}(s_V) = s_W,\ \forall W \subseteq V \right\}$$



Example: For $U = U_v \cup U_u$: $\mathcal{F}^+(U) = \{(s_{U_v}, s_{U_e}, s_{U_u}) \mid s_{U_e} = \mathcal{F}_{U_v,U_e}(s_v) = \mathcal{F}_{U_u,U_e}(s_u)\}$

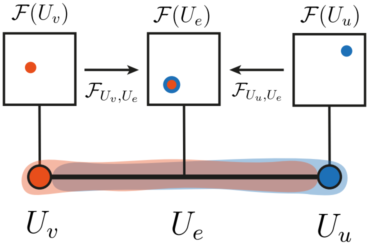
Sheaves on Graphs

Theorem

A $\mathcal{B}$-sheaf $\mathcal{F}$ on a space X uniquely induces a sheaf $\mathcal{F}^+$ on X such that $\mathcal{F}$ and $\mathcal{F}^+$ are canonically isomorphic.

Let $I(U) = \{V \mid V \in \mathcal{B}, V \subseteq U\}$. Then the sections of $\mathcal{F}^+$ are constructed as follows:

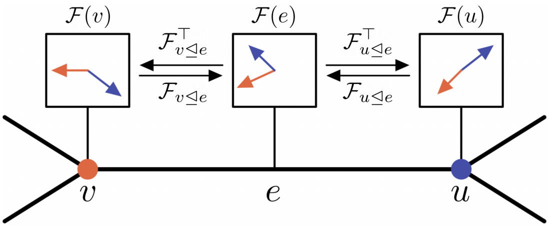
$$\mathcal{F}^+(U) := \left\{ (s_V)_{V \in I(U)} \in \prod_{V \in I(U)} \mathcal{F}(V) \mid \mathcal{F}_{V,W}(s_V) = s_W, \ \forall W \subseteq V \right\}$$



Example: For $U = U_v \cup U_u$: $\mathcal{F}^+(U) = \{(s_{U_v}, s_{U_e}, s_{U_u}) \mid s_{U_e} = \mathcal{F}_{U_v,U_e}(s_v) = \mathcal{F}_{U_u,U_e}(s_u)\}$

For each open $W \subseteq U$, $\mathcal{F}^+_{U,W}$ simply drops all the s_V, where $V \in I(U)$ but $V \notin I(W)$.

Cellular Sheaves

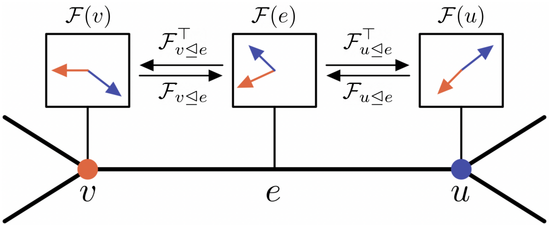


A *cellular sheaf*[6,7] $(G, \mathcal{F})$ of vector spaces on an undirected graph $G = (V, E)$ consists of:

1. A vector space $\mathcal{F}(v)$ for each $v \in V$.

[6] Curry, *Sheaves, cosheaves and applications*, 2014.
[7] Shepard, "A cellular description of the derived category of a stratified space", 1985.

Cellular Sheaves

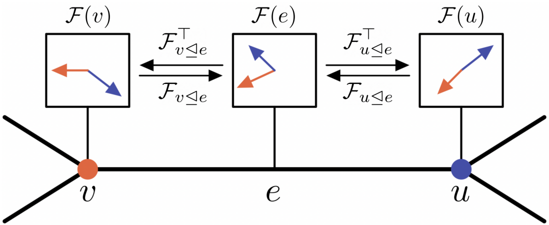


A *cellular sheaf*[6,7] $(G, \mathcal{F})$ of vector spaces on an undirected graph $G = (V, E)$ consists of:

1. A vector space $\mathcal{F}(v)$ for each $v \in V$.
2. A vector space $\mathcal{F}(e)$ for each $e \in E$.

[6] Curry, *Sheaves, cosheaves and applications*, 2014.
[7] Shepard, "A cellular description of the derived category of a stratified space", 1985.

Cellular Sheaves

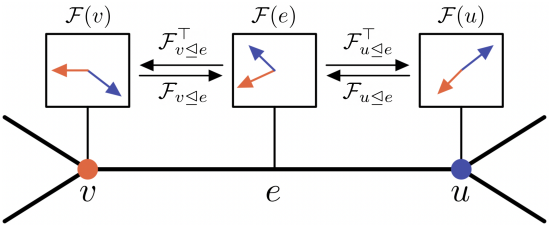


A *cellular sheaf*[6][7] $(G, \mathcal{F})$ of vector spaces on an undirected graph $G = (V, E)$ consists of:

1. A vector space $\mathcal{F}(v)$ for each $v \in V$.
2. A vector space $\mathcal{F}(e)$ for each $e \in E$.
3. A linear map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident $v \trianglelefteq e$ node-edge pair.

[6] Curry, *Sheaves, cosheaves and applications*, 2014.
[7] Shepard, "A cellular description of the derived category of a stratified space", 1985.

Cellular Sheaves



A *cellular sheaf*[6,7] $(G, \mathcal{F})$ of vector spaces on an undirected graph $G = (V, E)$ consists of:

1. A vector space $\mathcal{F}(v)$ for each $v \in V$.
2. A vector space $\mathcal{F}(e)$ for each $e \in E$.
3. A linear map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident $v \trianglelefteq e$ node-edge pair.

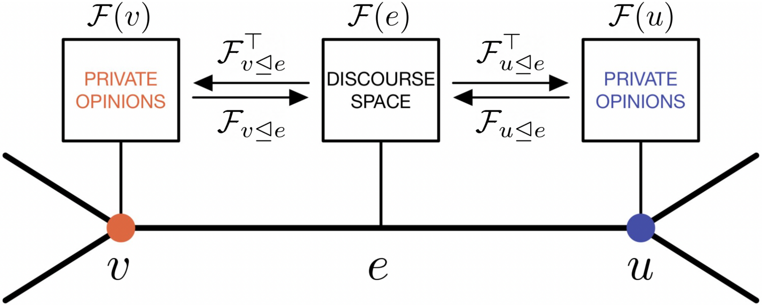
The vector spaces are called *stalks* and the linear maps are also known as *restriction maps*.

[6] Curry, *Sheaves, cosheaves and applications*, 2014.

[7] Shepard, "A cellular description of the derived category of a stratified space", 1985.

Opinion Dynamics

Opinion dynamics[8] provides a nice mental picture of cellular sheaves.



Nodes connected by edges represent people who are communicating. The vertex stalks contain the private opinions of the individuals, the edge stalks form a discourse space and the linear maps describe how the private opinions manifest publicly.

[8] Hansen and Ghrist, "Opinion dynamics on discourse sheaves", 2021.

Cochains and the coboundary operator

For a sheaf $(\mathcal{F}, G)$ we define the space of 0-*cochains* $C^0(G;\mathcal{F}) := \bigoplus_{v\in V} \mathcal{F}(v)$ and 1-*cochains* $C^1(G;\mathcal{F}) := \bigoplus_{e\in E} \mathcal{F}(e)$. This just gathers all the stalks into a vector space.

Cochains and the coboundary operator

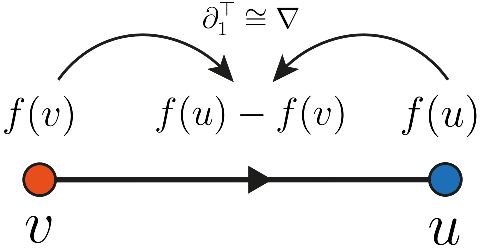
For a sheaf $(\mathcal{F}, G)$ we define the space of 0-*cochains* $C^0(G;\mathcal{F}) := \bigoplus_{v\in V}\mathcal{F}(v)$ and 1-*cochains* $C^1(G;\mathcal{F}) := \bigoplus_{e\in E}\mathcal{F}(e)$. This just gathers all the stalks into a vector space.

For some arbitrary choice of orientation for each edge $e = u \to v \in E$, define the linear coboundary map $\delta\colon C^0(G,\mathcal{F}) \to C^1(G,\mathcal{F})$ by $\delta(\mathbf{x})_e := \mathcal{F}_{v\trianglelefteq e}\mathbf{x}_v - \mathcal{F}_{u\trianglelefteq e}\mathbf{x}_u$.

Cochains and the coboundary operator

For a sheaf $(\mathcal{F}, G)$ we define the space of 0-*cochains* $C^0(G;\mathcal{F}) := \bigoplus_{v\in V}\mathcal{F}(v)$ and 1-*cochains* $C^1(G;\mathcal{F}) := \bigoplus_{e\in E}\mathcal{F}(e)$. This just gathers all the stalks into a vector space.

For some arbitrary choice of orientation for each edge $e = u \to v \in E$, define the linear coboundary map $\delta\colon C^0(G,\mathcal{F}) \to C^1(G,\mathcal{F})$ by $\delta(\mathbf{x})_e := \mathcal{F}_{v\trianglelefteq e}\mathbf{x}_v - \mathcal{F}_{u\trianglelefteq e}\mathbf{x}_u$.

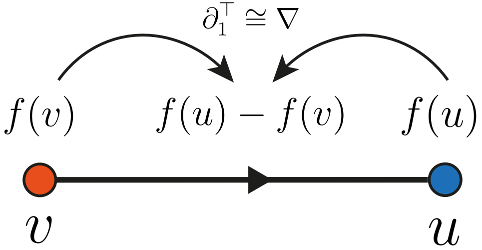


The transpose boundary operator behaves like a gradient. This is the coboundary of the sheaf with stalks $\mathbb{R}$ and identity restrictions.

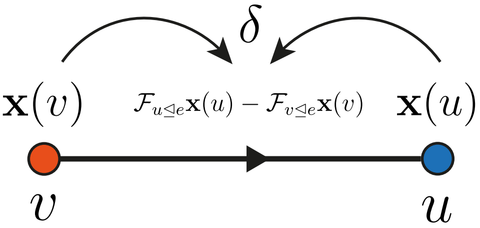
Cochains and the coboundary operator

For a sheaf $(\mathcal{F}, G)$ we define the space of 0-*cochains* $C^0(G;\mathcal{F}) := \bigoplus_{v\in V} \mathcal{F}(v)$ and 1-*cochains* $C^1(G;\mathcal{F}) := \bigoplus_{e\in E} \mathcal{F}(e)$. This just gathers all the stalks into a vector space.

For some arbitrary choice of orientation for each edge $e = u \to v \in E$, define the linear coboundary map $\delta\colon C^0(G,\mathcal{F}) \to C^1(G,\mathcal{F})$ by $\delta(\mathbf{x})_e := \mathcal{F}_{v\trianglelefteq e}\mathbf{x}_v - \mathcal{F}_{u\trianglelefteq e}\mathbf{x}_u$.



The transpose boundary operator behaves like a gradient. This is the coboundary of the sheaf with stalks $\mathbb{R}$ and identity restrictions.



The sheaf coboundary operator behaves like a covariant derivative.

The Dual Picture

In the previous talk, we saw how boundary operators ∂_k create a chain complex:

$$0 \to C_n(K, \mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K, \mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K, \mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K, \mathbb{R}) \xrightarrow{\partial_1} C_0(K, \mathbb{R}) \to 0$$

The Dual Picture

In the previous talk, we saw how boundary operators ∂_k create a chain complex:

$$0 \to C_n(K,\mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K,\mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K,\mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K,\mathbb{R}) \xrightarrow{\partial_1} C_0(K,\mathbb{R}) \to 0$$

Informally, we could have used the operators $\delta_{k-1} = \partial_k^\top$ to create a *cochain complex*:

$$0 \leftarrow C^n(K,\mathbb{R}) \leftarrow \cdots \xleftarrow{\delta_k} C^k(K,\mathbb{R}) \xleftarrow{\delta_2} C^{k-1}(K,\mathbb{R}) \cdots \xleftarrow{\delta_1} C^1(K,\mathbb{R}) \xleftarrow{\delta_0} C^0(K,\mathbb{R}) \leftarrow 0$$

The Dual Picture

In the previous talk, we saw how boundary operators ∂_k create a chain complex:

$$0 \to C_n(K,\mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K,\mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K,\mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K,\mathbb{R}) \xrightarrow{\partial_1} C_0(K,\mathbb{R}) \to 0$$

Informally, we could have used the operators $\delta_{k-1} = \partial_k^\top$ to create a *cochain complex*:

$$0 \leftarrow C^n(K,\mathbb{R}) \leftarrow \cdots \xleftarrow{\delta_k} C^k(K,\mathbb{R}) \xleftarrow{\delta_2} C^{k-1}(K,\mathbb{R}) \cdots \xleftarrow{\delta_1} C^1(K,\mathbb{R}) \xleftarrow{\delta_0} C^0(K,\mathbb{R}) \leftarrow 0$$

Similarly, we can define a *cohomology group* $H^k(K,\mathbb{R}) := \ker\, \delta_k / \mathrm{im}\, \delta_{k-1}$

The Dual Picture

In the previous talk, we saw how boundary operators ∂_k create a chain complex:

$$0 \to C_n(K, \mathbb{R}) \to \cdots \xrightarrow{\partial_{k+1}} C_k(K, \mathbb{R}) \xrightarrow{\partial_k} C_{k-1}(K, \mathbb{R}) \cdots \xrightarrow{\partial_2} C_1(K, \mathbb{R}) \xrightarrow{\partial_1} C_0(K, \mathbb{R}) \to 0$$

Informally, we could have used the operators $\delta_{k-1} = \partial_k^\top$ to create a *cochain complex*:

$$0 \leftarrow C^n(K, \mathbb{R}) \leftarrow \cdots \xleftarrow{\delta_k} C^k(K, \mathbb{R}) \xleftarrow{\delta_2} C^{k-1}(K, \mathbb{R}) \cdots \xleftarrow{\delta_1} C^1(K, \mathbb{R}) \xleftarrow{\delta_0} C^0(K, \mathbb{R}) \leftarrow 0$$

Similarly, we can define a *cohomology group* $H^k(K, \mathbb{R}) := \ker\, \delta_k / \mathrm{im}\, \delta_{k-1} \cong H_k(K, \mathbb{R})$

Sheaf Cohomology

Notice that the sheaf coboundary operator defines a *cochain complex* over the graph:

$$0 \leftarrow C^1(G;\mathcal{F}) \xleftarrow{\delta} C^0(G;\mathcal{F}) \leftarrow 0$$

Sheaf Cohomology

Notice that the sheaf coboundary operator defines a *cochain complex* over the graph:

$$0 \leftarrow C^1(G;\mathcal{F}) \overset{\delta}{\leftarrow} C^0(G;\mathcal{F}) \leftarrow 0$$

We can define the zero-th sheaf cohomology group as

$$H^0(G;\mathcal{F}) := \mathrm{ker}\ \delta_0/\mathrm{im}\ \delta_{-1} = \mathrm{ker}\ \delta_0$$

Sheaf Cohomology

Notice that the sheaf coboundary operator defines a *cochain complex* over the graph:

$$0 \leftarrow C^1(G;\mathcal{F}) \xleftarrow{\delta} C^0(G;\mathcal{F}) \leftarrow 0$$

We can define the zero-th sheaf cohomology group as

$$H^0(G;\mathcal{F}) := \ker\, \delta_0 / \mathrm{im}\, \delta_{-1} = \ker\, \delta_0$$

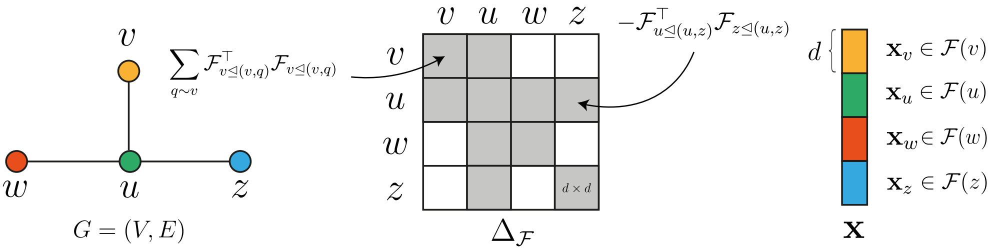
So what is the point?

The cohomology group $H^0(K,\mathbb{R}) \cong H_0(K,\mathbb{R})$ is boring. The sheaf structure gives us a much more interesting $H^0(G;\mathcal{F})$.

The Sheaf Laplacian

The *sheaf Laplacian*[9] is the linear operator $\delta^\top \delta$ and it is defined node-wise as

$$L_{\mathcal{F}}(\mathbf{x})_v := \sum_{v,u \trianglelefteq e} \mathcal{F}_{v \trianglelefteq e}^\top (\mathcal{F}_{v \trianglelefteq e} \mathbf{x}_v - \mathcal{F}_{u \trianglelefteq e} \mathbf{x}_u)$$

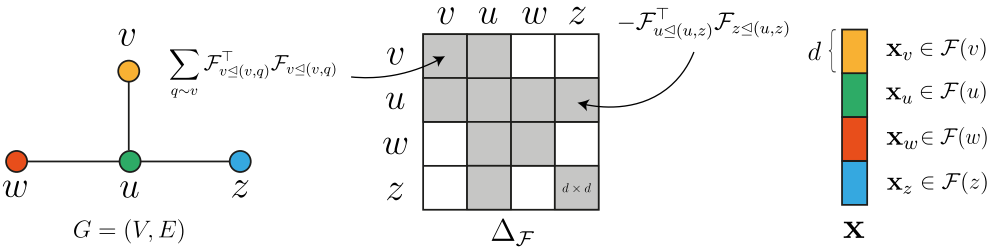


[9] Hansen and Ghrist, "Toward a spectral theory of cellular sheaves", 2019.

The Sheaf Laplacian

The *sheaf Laplacian*[9] is the linear operator $\delta^\top \delta$ and it is defined node-wise as

$$L_{\mathcal{F}}(\mathbf{x})_v := \sum_{v,u \trianglelefteq e} \mathcal{F}_{v \trianglelefteq e}^\top (\mathcal{F}_{v \trianglelefteq e} \mathbf{x}_v - \mathcal{F}_{u \trianglelefteq e} \mathbf{x}_u)$$

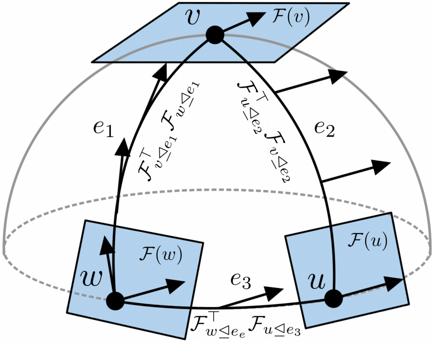


The normalised Laplacian $\Delta_{\mathcal{F}} := D^{-1/2} L_{\mathcal{F}} D^{-1/2}$, where D is the block-diagonal of $L_{\mathcal{F}}$. When $d = 1$ and $\mathcal{F}_{v \trianglelefteq e} = 1$, we obtain the (normalised) graph Laplacian.

[9] Hansen and Ghrist, "Toward a spectral theory of cellular sheaves", 2019.

Discrete Vector Bundles

The sheaves $(G, \mathcal{F})$ with orthogonal restriction maps are *discrete $O(d)$-bundles*[10].
Notice how easily we obtain some sort of geometric structure over the graph.



Analogy between parallel transport on a sphere and transport on a discrete vector bundle. A tangent vector is moved from $\mathcal{F}(w) \to \mathcal{F}(v) \to \mathcal{F}(u)$ and back.

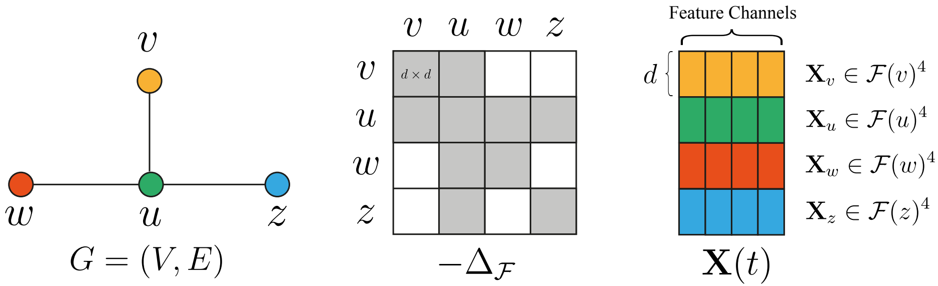
[10] Singer and Wu, "Vector diffusion maps and the connection Laplacian", 2012.

The Expressive Power of Sheaf Diffusion

Sheaf Diffusion

We now consider the *sheaf diffusion* process governed by the PDE:

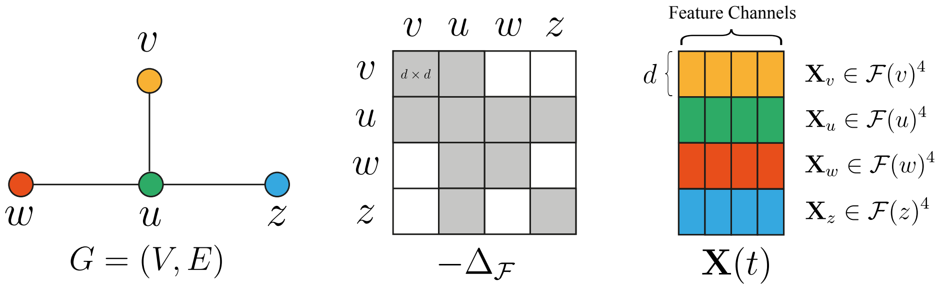
$$\mathbf{X}(0) = \mathbf{X}, \quad \dot{\mathbf{X}}(t) = -\Delta_{\mathcal{F}}\mathbf{X}(t)$$



Sheaf Diffusion

We now consider the *sheaf diffusion* process governed by the PDE:

$$\mathbf{X}(0) = \mathbf{X}, \quad \dot{\mathbf{X}}(t) = -\Delta_{\mathcal{F}}\mathbf{X}(t)$$

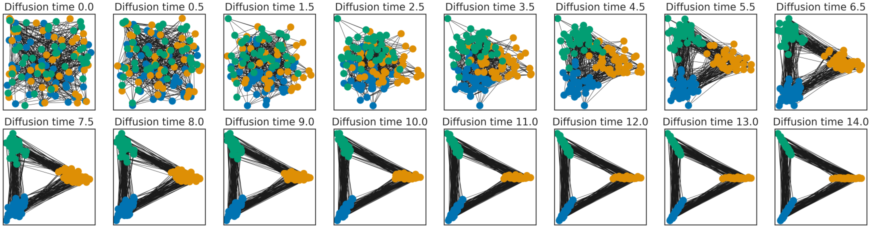


Theorem (Hodge Theorem)

As $t \to \infty$, the features converge to the projection of $\mathbf{X}(0)$ into $\ker \Delta_{\mathcal{F}} \cong H^0(G; \mathcal{F})$

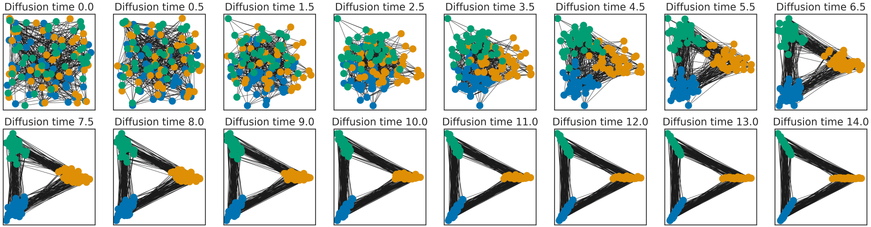
The Separation Power of Sheaf Diffusion

We want to look at what classes of sheaves can linearly separate the nodes of a graph in the infinite time limit of their diffusion process.



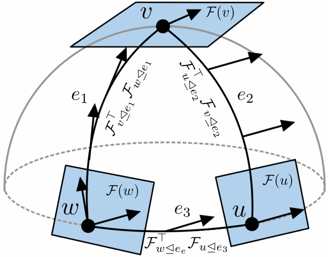
The Separation Power of Sheaf Diffusion

We want to look at what classes of sheaves can linearly separate the nodes of a graph in the infinite time limit of their diffusion process.



For that, we need to understand the properties of $\ker \Delta_{\mathcal{F}}$. In particular, when is $\ker \Delta_{\mathcal{F}} = 0$? We certainly do not want to converge to zero.

Harmonic Space of the Sheaf Laplacian

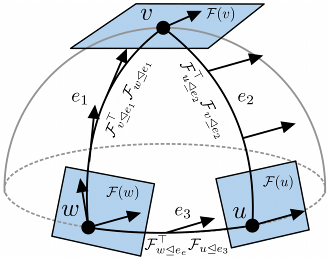


The transport is not path-independent because the vector returns in another position.

Let $(G, \mathcal{F})$ be a discrete $O(d)$-bundle. Given nodes $v, u \in V$ and a path $\gamma_{v\to u} = (v, v_1, \ldots, v_\ell, u)$ from v to u, we consider a notion of **transport** from the stalk $\mathcal{F}(v)$ to the stalk $\mathcal{F}(u)$ via map composition:

$$\mathbf{P}^{\gamma}_{v\to u} := (\mathcal{F}^\top_{u\trianglelefteq e}\mathcal{F}_{v_L\trianglelefteq e})\ldots(\mathcal{F}^\top_{v_1\trianglelefteq e}\mathcal{F}_{v\trianglelefteq e}) : \mathcal{F}(v) \to \mathcal{F}(u).$$

Harmonic Space of the Sheaf Laplacian



The transport is not path-independent because the vector returns in another position.

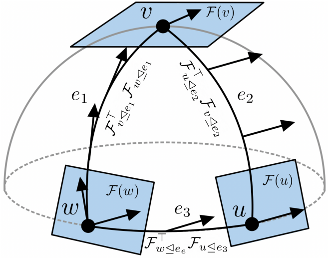
Let $(G, \mathcal{F})$ be a discrete $O(d)$-bundle. Given nodes $v, u \in V$ and a path $\gamma_{v\to u} = (v, v_1, \ldots, v_\ell, u)$ from v to u, we consider a notion of **transport** from the stalk $\mathcal{F}(v)$ to the stalk $\mathcal{F}(u)$ via map composition:

$$\mathbf{P}^\gamma_{v\to u} := (\mathcal{F}^\top_{u\trianglelefteq e}\mathcal{F}_{v_L\trianglelefteq e})\ldots(\mathcal{F}^\top_{v_1\trianglelefteq e}\mathcal{F}_{v\trianglelefteq e}) : \mathcal{F}(v) \to \mathcal{F}(u).$$

Main idea

The harmonic space of the Laplacian is related to the path-independence of the transport.

Harmonic Space of the Sheaf Laplacian



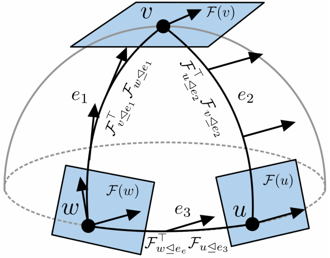
The transport is not path-independent because the vector returns in another position.

Let $\mathcal{F}$ be a discrete $O(d)$ bundle over a connected graph G with n nodes.

Proposition

Let $r := \max_{\gamma_{v \to u}, \gamma'_{v \to u}} ||\mathbf{P}^{\gamma}_{v \to u} - \mathbf{P}^{\gamma'}_{v \to u}||$, *then we have* $\lambda_0^{\mathcal{F}} \leq \frac{r^2}{2}$.

Harmonic Space of the Sheaf Laplacian



The transport is not path-independent because the vector returns in another position.

Let $\mathcal{F}$ be a discrete $O(d)$ bundle over a connected graph G with n nodes.

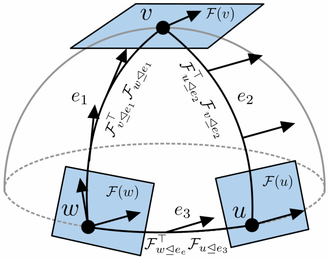
Proposition

Let $r := \max_{\gamma_{v\to u},\gamma'_{v\to u}} ||\mathbf{P}^{\gamma}_{v\to u} - \mathbf{P}^{\gamma'}_{v\to u}||$, *then we have* $\lambda_0^{\mathcal{F}} \leq \frac{r^2}{2}$.

Proposition

Let $||(\mathbf{P}^{\gamma}_{v\to v} - \mathbf{I})\mathbf{x}|| \geq \epsilon||\mathbf{x}||$ *for all cycles* $\gamma_{v\to v}$. *Then* $\lambda_0^{\mathcal{F}} \geq \epsilon^2 k_G$, *where* k_G *is a constant.*

Harmonic Space of the Sheaf Laplacian



The transport is not path-independent because the vector returns in another position.

Let $\mathcal{F}$ be a discrete $O(d)$ bundle over a connected graph G with n nodes.

Proposition

Let $r := \max_{\gamma_{v\to u}, \gamma'_{v \to u}} ||\mathbf{P}^{\gamma}_{v\to u} - \mathbf{P}^{\gamma'}_{v \to u}||$, *then we have* $\lambda_0^{\mathcal{F}} \leq \frac{r^2}{2}$.

Proposition

Let $||(\mathbf{P}^{\gamma}_{v \to v} - \mathbf{I})\mathbf{x}|| \geq \epsilon ||\mathbf{x}||$ *for all cycles* $\gamma_{v\to v}$. *Then* $\lambda_0^{\mathcal{F}} \geq \epsilon^2 k_G$, *where* k_G *is a constant.*

When the transport is path independent, $r = 0$ and $\epsilon = 0$ and so $\lambda_0^{\mathcal{F}} = 0$. This means there is at least one harmonic eigenvector and $\ker \Delta_{\mathcal{F}} \neq 0$.

Diffusion on Weighted Graphs

Consider the class of sheaves with stalks $\mathbb{R}$ and symmetric and non-zero scalar maps:

$$\mathcal{H}^1_{\text{sym}} := \{(\mathcal{F}, G) \mid \mathcal{F}(v) = \mathbb{R}, \mathcal{F}_{v \trianglelefteq e} = \mathcal{F}_{u \trianglelefteq e},\ \mathcal{F}_{v \trianglelefteq e} \neq 0, \forall v \in G\}$$

Diffusion on Weighted Graphs

Consider the class of sheaves with stalks $\mathbb{R}$ and symmetric and non-zero scalar maps:

$$\mathcal{H}^1_{\mathrm{sym}} := \{(\mathcal{F}, G) \mid \mathcal{F}(v) = \mathbb{R}, \mathcal{F}_{v \trianglelefteq e} = \mathcal{F}_{u \trianglelefteq e},\ \mathcal{F}_{v \trianglelefteq e} \neq 0, \forall v \in G\}$$

Proposition

Let $\mathcal{G}$ be the set of connected graphs $G = (V, E)$ with two classes $A, B \subset V$ such that for each $v \in A$, there exists $u \in A$ and an edge $(v, u) \in E$. Then $\mathcal{H}^1_{\mathrm{sym}}$ has linear separation power over $\mathcal{G}$.

Diffusion on Weighted Graphs

Consider the class of sheaves with stalks $\mathbb{R}$ and symmetric and non-zero scalar maps:

$$\mathcal{H}^1_{\mathrm{sym}} := \{(\mathcal{F}, G) \mid \mathcal{F}(v) = \mathbb{R}, \mathcal{F}_{v \trianglelefteq e} = \mathcal{F}_{u \trianglelefteq e},\ \mathcal{F}_{v \trianglelefteq e} \neq 0, \forall v \in G\}$$

Proposition

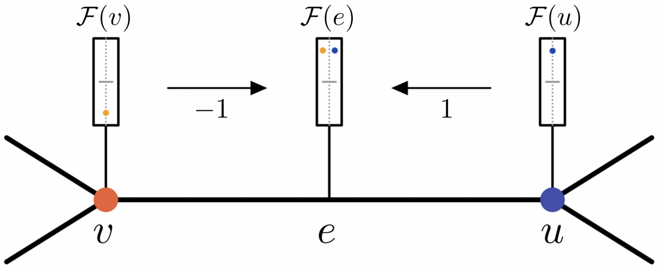
Let $\mathcal{G}$ be the set of connected graphs $G = (V, E)$ with two classes $A, B \subset V$ such that for each $v \in A$, there exists $u \in A$ and an edge $(v, u) \in E$. Then $\mathcal{H}^1_{\mathrm{sym}}$ has linear separation power over $\mathcal{G}$.

Proposition

Let $\mathcal{G}$ be the set of connected bipartite graphs $G = (A, B, E)$, with partitions A, B forming two classes and $|A| = |B|$. Then $\mathcal{H}^1_{\mathrm{sym}}$ cannot linearly separate any graph in $\mathcal{G}$ for any initial conditions $\mathbf{X}(0) \in \mathbb{R}^{n \times f}$.

Feature polarisation

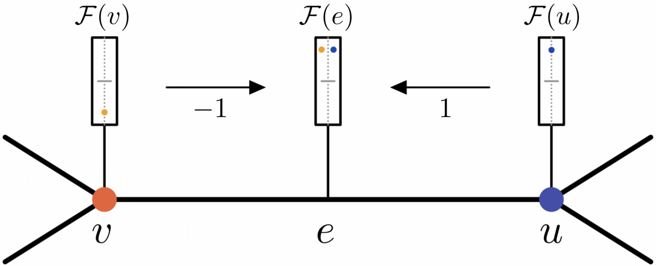
Let G be a conected graph with two classes A, B. Consider a sheaf with $\mathcal{F}_{v \trianglelefteq e} = -\alpha_e$ if $v \in A$ and $\mathcal{F}_{u \trianglelefteq e} = \alpha_e$ if $u \in B$ with $\alpha_e > 0$ for all $e \in E$.



Diffusion with opposite signs leads to feature polarisation.

Feature polarisation

Let G be a conected graph with two classes A, B. Consider a sheaf with $\mathcal{F}_{v \trianglelefteq e} = -\alpha_e$ if $v \in A$ and $\mathcal{F}_{u \trianglelefteq e} = \alpha_e$ if $u \in B$ with $\alpha_e > 0$ for all $e \in E$.



Diffusion with opposite signs leads to feature polarisation.

Proposition (Informal)

This type of sheaf can linearly separate the classes of any such graph for almost any initial conditions.

The blessing of dimensionality

Even with all this additional flexibility, dimension $d = 1$ still has a major limitation.

Proposition

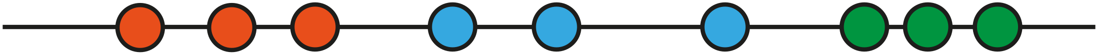
Let G be a connected graph with $C \geq 3$ classes. If $d = 1$, no sheaf can separate the classes for any $\mathbf{X}(0) \in \mathbb{R}^{n \times f}$.

The blessing of dimensionality

Even with all this additional flexibility, dimension $d = 1$ still has a major limitation.

Proposition

Let G be a connected graph with $C \geq 3$ classes. If $d = 1$, no sheaf can separate the classes for any $\mathbf{X}(0) \in \mathbb{R}^{n \times f}$.



This is a consequence of the fact the the features are projected on a subspace that is at most one-dimensional. In the best case, the classes are *pairwise* linearly separable.

Diagonal Restriction Maps

We can fix this by increasing the stalk dimension. Consider the class of sheaves with diagonal invertible maps and d-dimensional stalks:

$$\mathcal{H}^d_{\mathrm{diag}} := \{(\mathcal{F}, G) \mid \mathcal{F}_{v \trianglelefteq e} = \text{ invertible diagonal matrix}, \mathcal{F}(v) = \mathbb{R}^d\}$$

Diagonal Restriction Maps

We can fix this by increasing the stalk dimension. Consider the class of sheaves with diagonal invertible maps and d-dimensional stalks:

$$\mathcal{H}^d_{\mathrm{diag}} := \{(\mathcal{F}, G) \mid \mathcal{F}_{v \trianglelefteq e} = \text{ invertible diagonal matrix}, \mathcal{F}(v) = \mathbb{R}^d\}$$

Proposition

Let $\mathcal{G}$ be the set of connected graphs with nodes belonging to $C \geq 3$ classes. Then for $d \geq C$, $\mathcal{H}^d_{\mathrm{diag}}$ has linear separation power over $\mathcal{G}$.

Beyond diagonal maps

Diagonal restriction maps are extremely simple. Can we do better?

Beyond diagonal maps

Diagonal restriction maps are extremely simple. Can we do better?
We use the class of discrete $O(d)$-bundles $\mathcal{H}^d_{\mathrm{orth}} := \{(\mathcal{F}, G) \mid \mathcal{F}_{v \trianglelefteq e} \in O(d)\}$

Beyond diagonal maps

Diagonal restriction maps are extremely simple. Can we do better?
We use the class of discrete $O(d)$-bundles $\mathcal{H}^d_{\mathrm{orth}} := \{(\mathcal{F}, G) \mid \mathcal{F}_{v \trianglelefteq e} \in O(d)\}$

Theorem

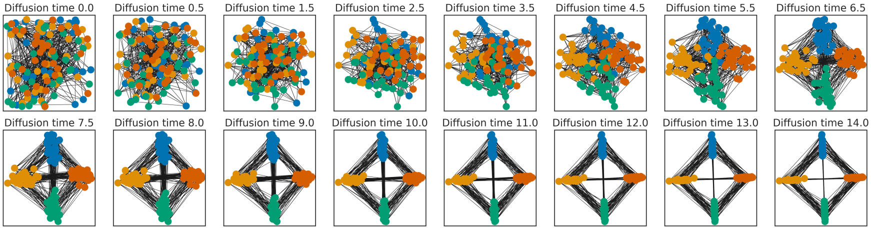
Let $\mathcal{G}$ be the class of connected graphs with $C \leq 2d$ classes. Then, for all $d \in \{2, 4\}$, $\mathcal{H}^d_{\mathrm{orth}}$ has linear separation power over $\mathcal{G}$.

Beyond diagonal maps

Diagonal restriction maps are extremely simple. Can we do better?
We use the class of discrete $O(d)$-bundles $\mathcal{H}^d_{\mathrm{orth}} := \{(\mathcal{F}, G) \mid \mathcal{F}_{v \trianglelefteq e} \in O(d)\}$

Theorem

Let $\mathcal{G}$ be the class of connected graphs with $C \leq 2d$ classes. Then, for all $d \in \{2, 4\}$, $\mathcal{H}^d_{\mathrm{orth}}$ has linear separation power over $\mathcal{G}$.



The Power of Sheaf Diffusion: Review

Takeaway

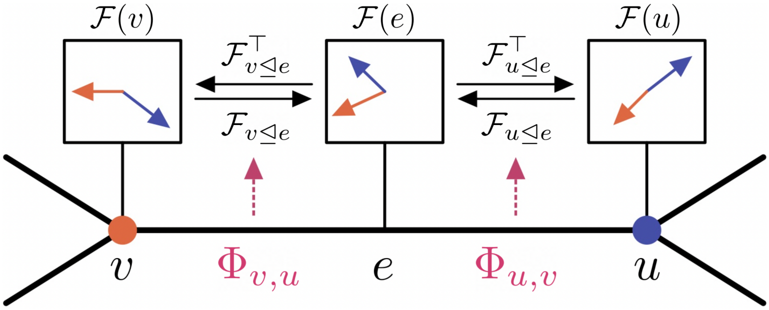
Different classes of sheaves induce diffusion processes with different capabilities. Furthermore, any node-classification problem can be reduced to performing diffusion with the right sheaf.

Learning Sheaves

Learning sheaves

Each $d \times d$ matrix $\mathcal{F}_{v \trianglelefteq e}$ is learned via a parametric function $\Phi : \mathbb{R}^{d \times 2} \to \mathbb{R}^{d \times d}$:

$$\mathcal{F}_{v \trianglelefteq e := (v,u)} = \Phi(\mathbf{x}_v, \mathbf{x}_u) \tag{3}$$



The restriction maps are learned from data.

Time-evolving sheaves[11]

We want to learn a sheaf from the latest available features.

$$\dot{\mathbf{X}}(t) = -\sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I}_n \otimes \mathbf{W}_1)\mathbf{X}(t)\mathbf{W}_2\Big), \tag{4}$$

[11] A sheaf convolutional model with hand-crafted sheaves was originally proposed by Hansen and Gebhart, "Sheaf Neural Networks", 2020

Time-evolving sheaves[11]

We want to learn a sheaf from the latest available features.

$$\dot{\mathbf{X}}(t) = -\sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I}_n \otimes \mathbf{W}_1)\mathbf{X}(t)\mathbf{W}_2\Big), \tag{4}$$

We also consider a discrete version of this equation, with different weights at each layer t.

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I} \otimes \mathbf{W}_1^t)\mathbf{X}_t\mathbf{W}_2^t\Big), \tag{5}$$

[11] A sheaf convolutional model with hand-crafted sheaves was originally proposed by Hansen and Gebhart, "Sheaf Neural Networks", 2020

Time-evolving sheaves[11]

We want to learn a sheaf from the latest available features.

$$\dot{\mathbf{X}}(t) = -\sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I}_n \otimes \mathbf{W}_1)\mathbf{X}(t)\mathbf{W}_2\Big), \tag{4}$$

We also consider a discrete version of this equation, with different weights at each layer t.

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I} \otimes \mathbf{W}_1^t)\mathbf{X}_t\mathbf{W}_2^t\Big), \tag{5}$$

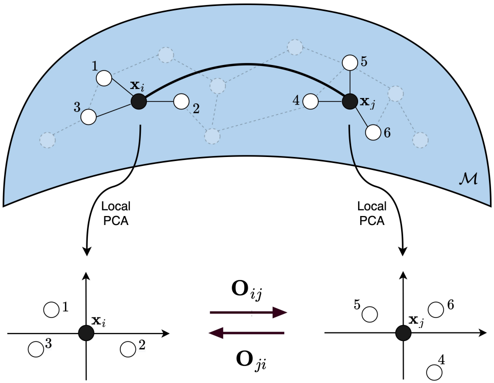
Main idea

The sheaf evolves over time as a function of the data $(G, \mathcal{F}(t)) = g(G, \mathbf{X}(t); \theta)$.

[11] A sheaf convolutional model with hand-crafted sheaves was originally proposed by Hansen and Gebhart, "Sheaf Neural Networks", 2020

Sheaves at pre-processing time

In a recent paper[12] we showed how by assuming that the graph is sampled from a manifold, we can construct a reasonable sheaf at pre-processing time.



We adapt existent methods[13] to learn the connection that best aligns the tangent spaces of the nodes.

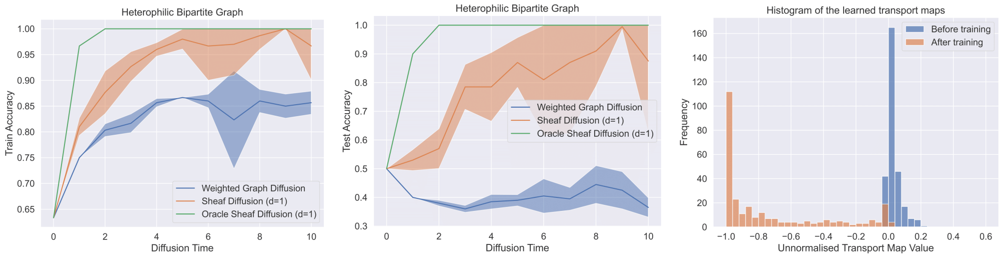
[12] Barbero et al., "Sheaf Neural Networks with Connection Laplacians", 2022.

[13] Singer and Wu, "Vector diffusion maps and the connection Laplacian", 2012.

Results

Synthetic Experiment: Opinion Polarisation

We have a bipartite graph with equally sized partitions that we try to distinguish. $\mathbf{X}(0)$ is not linearly separable. We use a simple sheaf diffusion process with a learned sheaf Laplacian (i.e. no weights and non-linearities)



Training (*Left*) and Testing (*Middle*) accuracy as a function of diffusion time. Learned sheaf Laplacian for $t >> 0$. (*Right*)

Real-World Evaluation

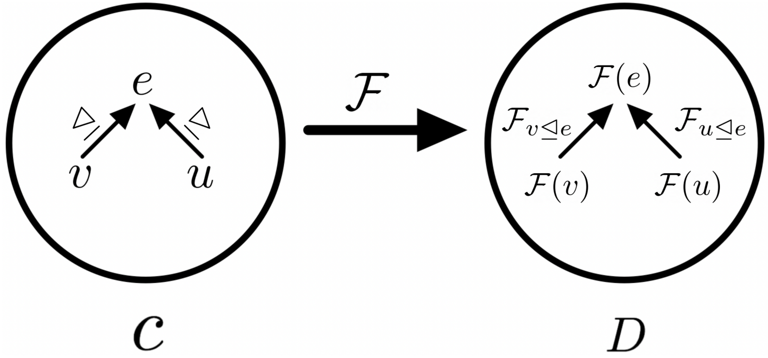
We evaluate on multiple node-classifications tasks with various degrees of homophily[14].

	Texas	**Wisconsin**	**Film**	**Squirrel**	**Chameleon**	**Cornell**	**Citeseer**	**Pubmed**	**Cora**
Hom level	**0.11**	**0.21**	**0.22**	**0.22**	**0.23**	**0.30**	**0.74**	**0.80**	**0.81**
#Nodes	183	251	7,600	5,201	2,277	183	3,327	18,717	2,708
#Edges	295	466	26,752	198,493	31,421	280	4,676	44,327	5,278
#Classes	5	5	5	5	5	5	7	3	6
Diag-NSD	**85.67**±6.95	**88.63**±2.75	**37.79**±1.01	**54.78**±1.81	**68.68**±1.73	**86.49**±7.35	**77.14**±1.85	89.42±0.43	87.14±1.06
O(d)-NSD	**85.95**±5.51	**89.41**±4.74	**37.81**±1.15	**56.34**±1.32	**68.04**±1.58	**84.86**±4.71	76.70±1.57	**89.49**±0.40	86.90±1.13
Gen-NSD	82.97±5.13	**89.21**±3.84	**37.80**±1.22	53.17±1.31	67.93±1.58	**85.68**±6.51	76.32±1.65	89.33±0.35	87.30±1.15
GGCN	**84.86**±4.55	86.86±3.29	37.54±1.56	**55.17**±1.58	**71.14**±1.84	**85.68**±6.63	**77.14**±1.45	89.15±0.37	**87.95**±1.05
H2GCN	**84.86**±7.23	87.65±4.98	35.70±1.00	36.48±1.86	60.11±2.15	82.70±5.28	77.11±1.57	**89.49**±0.38	**87.87**±1.20
GPRGNN	78.38±4.36	82.94±4.21	34.63±1.22	31.61±1.24	46.58±1.71	80.27±8.11	77.13±1.67	87.54±0.38	**87.95**±1.18
FAGCN	82.43±6.89	82.94±7.95	34.87±1.25	42.59±0.79	55.22±3.19	79.19±9.79	N/A	N/A	N/A
MixHop	77.84±7.73	75.88±4.90	32.22±2.34	43.80±1.48	60.50±2.53	73.51±6.34	76.26±1.33	85.31±0.61	87.61±0.85
GCNII	77.57±3.83	80.39±3.40	37.44±1.30	38.47±1.58	63.86±3.04	77.86±3.79	**77.33**±1.48	**90.15**±0.43	**88.37**±1.25
Geom-GCN	66.76±2.72	64.51±3.66	31.59±1.15	38.15±0.92	60.00±2.81	60.54±3.67	**78.02**±1.15	**89.95**±0.47	85.35±1.57
PairNorm	60.27±4.34	48.43±6.14	27.40±1.24	50.44±2.04	62.74±2.82	58.92±3.15	73.59±1.47	87.53±0.44	85.79±1.01
GraphSAGE	82.43±6.14	81.18±5.56	34.23±0.99	41.61±0.74	58.73±1.68	75.95±5.01	76.04±1.30	88.45±0.50	86.90±1.04
GCN	55.14±5.16	51.76±3.06	27.32±1.10	53.43±2.01	64.82±2.24	60.54±5.30	76.50±1.36	88.42±0.50	86.98±1.27
GAT	52.16±6.63	49.41±4.09	27.44±0.89	40.72±1.55	60.26±2.50	61.89±5.05	76.55±1.23	87.30±1.10	86.33±0.48
MLP	80.81±4.75	85.29±3.31	36.53±0.70	28.77±1.56	46.21±2.99	81.89±6.40	74.02±1.90	75.69±2.00	87.16±0.37

[14] Rozemberczki et al., "Multi-scale attributed node embedding", 2021; Pei et al., "Geom-gcn: Geometric graph convolutional networks", 2020.

The Bigger Picture: A Category Theory Perspective

From the perspective of category theory, cellular sheaves are *functors* from the category describing the incidence structure of a graph to some other category.



Collaborators

Many of the ideas and results presented in these talks are the results of many collaborations and interactions with:

- Fabrizio Frasca (Twitter)
- Francesco di Giovanni (Twitter)
- Federico Barbero (University of Cambridge)
- Yu Guang Wang (Shanghai Jiao Tong University)
- Guido Montufar (UCLA & Max Planck Institute for Mathematics in the Sciences)
- Nina Otter (Queen Mary University of London)
- Ben Chamberlain (Twitter)
- Michael Bronstein (Twitter & University of Oxford)
- Pietro Liò (University of Cambridge)

Thank you for your attention!

Email: `cb2015@cam.ac.uk`
Twitter: `@crisbodnar`