Data Structures as Sheaves

Michael Robinson

© 2015 Michael Robinson
This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
Mathematical dependency tree

- Sheaf cohomology (Lectures 7, 8)
  - Cellular sheaves (Lectures 3, 4)
    - CW complexes
      - Simplicial Complexes
        - Abstract Simplicial Complexes

- Homology
  - Linear algebra (Lectures 5, 6)
  - Manifolds
    - Calculus
      - Topology
        - Lecture 2

- Sheaves

- de Rham cohomology (Stokes' theorem)

- Set theory

- Michael Robinson
Session objectives

- How do sheaves extend well-known data structures?
- How do I translate between sheaf-based data structures?
- What can I do once I have a sheaf?
What is a sheaf?
What is a sheaf?
If labels on the graph are not systematically related to one another...

Consistency checks become *ad hoc*

Cross-modality inference is no longer possible
<table>
<thead>
<tr>
<th>Vertex weighted → sheaves</th>
<th>Hyperedge-weighted → cosheaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Vertex has nontrivial stalk</td>
<td>• Toplex has nontrivial stalk</td>
</tr>
<tr>
<td>• All restrictions are zero maps</td>
<td>• All extensions are zero maps</td>
</tr>
<tr>
<td>• The resulting sheaf is flabby</td>
<td>• The resulting cosheaf is coflabby*</td>
</tr>
</tbody>
</table>

* This is such a fun term – but I don't see it much in use
Flow sheaves

- Start with a collection of paths along which material flows
- Label each track segment with amount of material on that segment
Flow sheaves

- Conservation law enforced at each vertex
- Depending on precisely how we count material (in $\mathbb{N}$ or $\mathbb{R}$, for instance), we might write the conservation law as

\[
a + b = c + d \text{ or } a + b - c - d = 0
\]

\[
d + e = f
\]
Flow sheaf

- Each degree $n$ vertex is assigned a free $sR$-module* of rank $(n - 1)$ for material measured in a semiring $sR$
- Restriction maps are projections

* or just a vector space if it seems easier
Local consistency

A flow sheaves encodes a notion of consistency between adjacent faces.

Cannot label this edge in a way consistent with the data.

It's an indication that a flow was incorrectly measured somewhere in this vicinity.
Inferential ambiguity

Depending on how we make measurements, we might not get “the full story” of the flow

Inconsistent: no inference possible

Exactly one inference

Many possible inferences
Bayesian networks
Probability spaces

Start with a set of random variables $X_0, X_1, \ldots, X_n$

Consider the set $P(X_0, X_1, \ldots, X_n)$ of all joint probability distributions over these random variables

- These are the nonnegative measures (generalized functions)

$$f = f(X_0, X_1, \ldots, X_n)$$

with unit integral

- This is not a vector space – adding probability distributions doesn't yield another distribution
Marginalization cosheaf

There is a natural map

\[ P(X_0, X_1, \ldots, X_n) \to P(X_0, X_1, \ldots, X_{n-1}) \]

via *marginalization*, namely

\[ f(X_0, X_1, \ldots, X_{n-1}) = \int f(X_0, X_1, \ldots, X_n) \, dX_n \]

- There similar maps for marginalizing out the other random variables, too
- This yields a cosheaf on the complete $n$-simplex!
Bayes' rule

Conditional probabilities produce maps going the other way... For instance,

\[ P(X_0, X_1, \ldots, X_{n-1}) \rightarrow P(X_0, X_1, \ldots, X_n) \]

is parameterized by functions \( C \)

\[ F(X_0, X_1, \ldots, X_n) = C(X_0, X_1, \ldots, X_n) f(X_0, X_1, \ldots, X_{n-1}) \]

usually, one writes the arguments to \( C \) like

\[ C = C(X_n \mid X_0, X_1, \ldots, X_{n-1}) \]

So... conditional probabilities yield a sheaf on part of the \( n \)-simplex
Small Bayes net example

- Consider two binary random variables $X$ and $Y$ with a given conditional $C(Y \mid X)$

$$P(X) \xrightarrow{(1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1)} P(X,Y) \xrightarrow{(1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1)} P(Y)$$

Marginalization cosheaf

$$P(X) \xrightarrow{\begin{pmatrix} p(0|0) & 0 \\ p(1|0) & 0 \\ 0 & p(0|1) \\ 0 & p(1|1) \end{pmatrix}} P(X,Y) = C(Y \mid X) \xrightarrow{P(Y)}$$

Conditional sheaf
Linear translation-invariant filters
How does a sheaf model a signal?

The sheaf of continuous functions

Simplicial complex for $\mathbb{R}$

$C((-2,0),\mathbb{R})$

$C((-1,0),\mathbb{R})$

$C((-1,1),\mathbb{R})$

$C((0,1),\mathbb{R})$

$C((0,2),\mathbb{R})$
How does a sheaf model a signal?

Set of real-valued continuous functions on \((-2,0)\)

Restriction map "chops" off portion of a function

Space of global sections is the set of continuous functions on \(\mathbb{R}\)
A sheaf morphism ...

- ... takes data in the stalks of two sheaves ...

\[
\begin{align*}
\begin{pmatrix}
1 & 1 \\
2 & 0
\end{pmatrix} & \rightarrow \mathbb{R}^2 \\
& \rightarrow \begin{pmatrix}
2 & 0 \\
1 & 1
\end{pmatrix} \\
\begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 0
\end{pmatrix} & \rightarrow \mathbb{R}^3 \\
& \rightarrow \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix} \\
& \rightarrow \mathbb{R}^2
\end{align*}
\]
A sheaf morphism ...

... and relates them through linear maps ...

\[
\begin{pmatrix}
\frac{1}{2} & 0 \\
-\frac{1}{2} & -1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & -1 \\
1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 \\
0 & -1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 \\
-1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 \\
-1 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 \\
-1 & 0
\end{pmatrix}
\]
**A sheaf morphism ...**

- ... so the diagram commutes!
A sampling morphism is ...

\[ C((-2,0), \mathbb{R}) \rightarrow \mathbb{R} \]
\[ C((-1,0), \mathbb{R}) \rightarrow 0 \]
\[ C((-1,1), \mathbb{R}) \rightarrow \mathbb{R} \]
\[ C((0,1), \mathbb{R}) \rightarrow 0 \]
\[ C((0,2), \mathbb{R}) \rightarrow \mathbb{R} \]

Evaluate at -1
Evaluate at 0
Evaluate at 1
An ambiguity sheaf is ...

- The collection of kernels of these sampling maps

\[ Z_{-1}(-2,0) \rightarrow C((-2,0), \mathbb{R}) \rightarrow \mathbb{R} \]
\[ \downarrow \quad \downarrow \quad \downarrow \]
\[ C((-1,0), \mathbb{R}) \rightarrow C((-1,0), \mathbb{R}) \rightarrow 0 \]
\[ \downarrow \quad \downarrow \]
\[ Z_0(-1,1) \rightarrow C((-1,1), \mathbb{R}) \rightarrow \mathbb{R} \]
\[ \downarrow \quad \downarrow \]
\[ C((0,1), \mathbb{R}) \rightarrow C((0,1), \mathbb{R}) \rightarrow 0 \]
\[ \downarrow \quad \downarrow \]
\[ Z_1(0,2) \rightarrow C((0,2), \mathbb{R}) \rightarrow \mathbb{R} \]

The set of continuous functions on (-2,0) that vanish at -1.

Evaluate at -1

Evaluate at 0

Evaluate at 1
Sections of the ambiguity sheaf are functions that appear to be all zero under the sampling.
The general sampling theorem

- Given a sheaf $S$ and a sampling of it, construct the ambiguity sheaf $A$

**Theorem:**
- Perfect reconstruction of global sections of $S$ is possible if and only if the only global section of $A$ is the zero function

- “No ambiguities means it's possible to reconstruct”
Nyquist-Shannon sampling

- Encode signals as a sheaf of bandlimited functions $BF$ over $\mathbb{R}$, with bandwidth $B$
- It's easier to work in the frequency domain:
  \[ BF = \{ f \in C(\mathbb{R}, \mathbb{C}) | \text{supp } f \subseteq [-B, B] \} \]
- Samples are taken at integers, obtained by inverse Fourier transform
- For instance at $n$, we sample using the function $M^n$
  \[ M_n(f) = \int_{-B}^{B} f e^{-2n\pi ix} \, dx \]
Nyquist-Shannon sampling

- The ambiguity sheaf identifies bandlimited functions with zeros at specific locations

\[ A_n = \{ f \in BF \mid M_n(f) = 0 \} \]

\[ = \{ \text{Bandlimited functions that are zero at } n \} \]
Nyquist-Shannon sampling

- The ambiguity sheaf identifies bandlimited functions with zeros at specific locations

\[ A_n = \{ f \in BF \mid M_n(f) = 0 \} \]

= \{ Bandlimited functions that are zero at \( n \) \}

Global sections of the ambiguity sheaf are bandlimited functions that vanish at every integer

There aren't any if \( B < 1/2 \)
Filters as sheaf morphisms

- **Theorem**: Every discrete-time LTI filter can be encoded as a sequence of two sheaf morphisms

\[ S_1 \xrightarrow{} S_2 \xrightarrow{} S_3 \]

**Sheaf formalism**

---

**Hardware**

Shift register

Weighted sum

Input ——— Internal state ——— Output
Input sheaf

- Sections of this sheaf are timeseries, instead of continuous functions
Output sheaf

- The output sheaf is the same
The internal state

- Contents of the shift register at each timestep
- \( N = 3 \) shown

\[
\begin{align*}
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{pmatrix} & \quad \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{pmatrix} & \quad \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{pmatrix} \\
\begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix} & \quad \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix} & \quad \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix} \\
\begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix} & \quad \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix} & \quad \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\end{align*}
\]
The internal state

- Loads a new value with each timestep
The internal state

- Produces average of the shift register at each timestep
Finishing both morphisms

• Put in a few zero maps!
A single timestep

- Sections of the sheaves linked together give possible input/output combinations of the filter.
Of course, this extends...

- Sections of the sheaves linked together give possible input/output combinations of the filter.
How this formalism helps...

• Of course, it corresponds nicely to the hardware
  … BUT...

• It's easy to splice in nonlinear operations
• It works on nontrivial base spaces: A systematic study of filtering is now possible on cell complexes

Sheaves make it easy to invent topological filters that have controlled performance characteristics
Filtering sinusoids from noise

Signals at ports of standard LPF filter

Time

Signal

Input

Output
Filtering out chirpy signals

Loss of signal as frequency changes
Phase reversal
Filtering out chirpy signals

Signals at ports of variable-bandwidth LPF filter

Unstable amplitude
Filtering out chirpy signals

Poor initial estimate of frequency

Loss of amplitude – filter cannot “keep up”!
Circumventing bandwidth limits

• More averaging in a connected window leads to:
  – More noise cancellation (Good)
  – More distortion to the signal (Bad)

• Can **safely do more** averaging by collecting samples at “similar places” across the **entire** signal
Filter block diagram

Stage 1: Topological estimation
- Point cloud metric construction
- Grouping sheaf construction
- Average along rows

Stage 2: Topological filtering
- Averaging filter
- Output signal
Stage 1: Topological estimation

Image spatial domain

Pixel neighborhood

Point cloud in $\mathbb{R}^{16}$
Stage 2: Grouping sheaf

Distance matrix of point cloud

Locations of nearest neighbors

Values of the signal at the neighbors

Sample number

Sample number

Sample number

Average

Process execution
Topological filter results

- Extremely stable output amplitude

Some low frequency distortion
Compare: standard adaptive filter
Input image
Fixed frequency image filter
Topological filter output
Error contributions

Topological filter stage-wise performance at 2.5x filter BW

Matching Dimension: 20

Boxcar filter

Using topological estimator

Using true topology

Topological estimation error

Error

0.7

0.6

0.5

0.4

0.3

0.2

0.1

0

Noise

0

0.2

0.4

0.6

0.8

1
Error contributions

Topological filter stage-wise performance at 2.5x filter BW

Matching
Dimension: 25

Boxcar filter

Using topological estimator

Using true topology
Error contributions

Topological filter stage-wise performance at 2.5x filter BW

Matching
Dimension: 30

Boxcar filter
Using topological estimator
Using true topology

Noise
Next up...

- Informal social
- Next lecture (tomorrow morning): Categorification and Chain Complexes
Further reading...


