Sheaves as a Framework for Understanding and Interpreting Model Fit

Henry Kvinge, Brett Jefferson, Cliff Joslyn, Emilie Purvine
Pacific Northwest National Laboratory
Seattle, WA, USA
first.last@pnnl.gov

Abstract

As data grows in size and complexity, finding frameworks which aid in interpretation and analysis has become critical. This is particularly true when data comes from complex systems where extensive structure is available, but must be drawn from peripheral sources. In this paper we argue that in such situations, sheaves can provide a natural framework to analyze how well a statistical model fits at the local level (that is, on subsets of related datapoints) vs the global level (on all the data). The sheaf-based approach that we propose is suitably general enough to be useful in a range of applications, from analyzing sensor networks to understanding the feature space of a deep learning model.

1. Introduction

Data is being collected at an ever-increasing rate in an ever-broader range of modalities. It is more and more frequently the case that extracting useful information from such large datasets requires the integration of sophisticated analytical techniques in combination with deep domain knowledge (often drawn from independent databases). While progress has been made in developing approaches to better visualize and explore the data itself, techniques for bringing outside knowledge into the analysis of the statistical models built on top of this data either remain mostly ad hoc or have not caught up to the size or scope of current state-of-the-art models. In this paper we propose a sheaf-theoretic approach to address this challenge.

Despite their ubiquity in mathematics, sheaves have only recently started playing a role in data science, where they have been leveraged as data structures which can systematically capture information from many non-independent data-streams. Sheaf frameworks have been developed for uncertainty quantification in geolocation [5], air traffic control monitoring [8], and learning signals on graphs [2]. Much of the inspiration for the present work comes from sheaf-theoretic constructions meant to analyze sensor networks [11], [1]. We use these ideas to analyze the fit of data-driven models.

Our motivation arises from the observation that the quality of a model's "fit" may vary dramatically across subpopulations of a dataset. Indeed, it is increasingly apparent that the construction of models that are both robust and often highly-overparametrized requires understanding model performance not only at the global level (for example, the accuracy over an entire dataset) but also at the local level (precision on meaningful subsets of the data) [14]. While our focus is not on overparametrized models in this work, we do give an example of how our framework might be applied to this setting in Section 3.1.

To define sheaves on a dataset we first construct a topology based on metadata, with open sets corresponding to subsets of related points. We build a sheaf and presheaf on top of this topology. We call the sheaf the *data sheaf* $\mathcal{D}$. It consists of all possible data value observations. Then the *model presheaf* $\mathcal{M}$ consists of a specified family of functions associated with each open set. For example, each data point might be indexed by a spatial location on Earth, an open set U might then consist of spatial locations that are nearby each other, and a data *observation* might be a measurement of temperature and wind speed. Then a section of the data sheaf at U is a function $f : U \to \mathbb{R}^2$ which associates an element of $\mathbb{R}^2$ (temperature and wind speed) to each spatial location in U. If we wish to predict wind speed from temperature, we might choose $\mathcal{M}(U)$ to consist of all 1-dimensional affine subspaces (lines) in $\mathbb{R}^2$ that relate these two quantities. The process of modeling data associated with the points in U as a line in $\mathbb{R}^2$ is equivalent to defining a map from the space of sections $\mathcal{D}(U)$ to the space of sections $\mathcal{M}(U)$.

As suggested by this example, we identify a method of modeling data on each open set U in our topology with a map $\Phi_U^{\mathcal{M}} : \mathcal{D}(U) \to \mathcal{M}(U)$. That is, for an observation of data on U we have a rule for picking a model. In general, $\Phi^{\mathcal{M}}$ will not be a presheaf morphism. Inspired by the notion of consistency from [11], we introduce a family of statistics, model map *inconsistency*, which take values in $\mathbb{R}_{\geq 0}$ and measure the extent to which $\Phi^{\mathcal{M}}$ is not a presheaf morphism. Indeed, we show in Proposition 3.1 that $\Phi^{\mathcal{M}}$ is a presheaf

Researcher	Publications
a	5
b	6
c	8
d	7
e	4
f	5

Table 1. Number of publications per researcher in Example 1.1

morphism if and only if the inconsistency of $\Phi^{\mathcal{M}}$ is 0.

Model map inconsistency is important because it allows us to point to specific subpopulations of a dataset on which a given model's ability to fit the data changes. This is critical because in the real world, the summary statistics used in academic benchmark studies often do not provide sufficient feedback on model behavior and performance. Indeed, we need to understand specific, systemic failures in a model before it is deployed. Further, note that our presheaf structure can also capture specific statistics associated with a global model. The simplest example of this might be the accuracy of a predictive model across various subpopulations of a test set. In this case "model" in the term model presheaf, might more appropriately be called "model performance presheaf". To illustrate this latter use, we end this work by using inconsistency statistics to probe the feature space of a ResNet50 convolutional neural network [3] that has been trained on the large image database ImageNet [12]. We show how our sheaf-theoretic framework can be used to illuminate biases in rich and complex computer vision models such as this.

2. Datasets and models as sheaves

2.1. Notation and underlying topology

Below we assume that set I indexes the elements of the dataset D that we will be working with so that

$$D = \{x_i \mid i \in I\}, \tag{1}$$

and each $x_i \in D$ takes a value in *target space* Y. We could equivalently encode D as a function $f_D : I \to Y$, where $f_D(i) = x_i$. This interpretation will be useful when defining data sheaves in Section 2.2.

Example 1. *1. (Toy example) Let D be a dataset that consists of the number of publications for* 6 *researchers. Denote these researchers by a, b, c, d, e, f. The number of publications of each is recorded in Table 1. Then $I = \{a, b, c, d, e, f\}$ and $Y = \mathbb{R}$ (we use $\mathbb{R}$ rather than $\mathbb{N}$ because we will later want to take averages in the target space).*

2. *(Sensors) Suppose that D consists of real-valued readings from a collection of distinct sensors. Then I indexes the set of all sensors, $Y = \mathbb{R}$, and $f_D : I \to \mathbb{R}$ is a function that assigns to each sensor its reading.*

3. *(Gene expression) If D is a gene expression level dataset where readings are taken from each gene at a fixed set of r time steps, then I is a set that indexes all genes whose expression level are being measured, $Y = \mathbb{R}^r$, and $f_D : I \to \mathbb{R}^r$ assigns to each gene a vector recording its r readings.*

4. *(Computer vision feature extractor) Let $J \subset \mathbb{R}^{h \times w \times 3}$ be a collection of height h and width w RGB images and let φ be a convolutional neural network (CNN) feature extractor that has been trained to map images from their usual pixel space, to vectors in a feature space $\mathbb{R}^r$ where spatial relationships can be related to image content (that is, if $x_1, x_2 \in \mathbb{R}^{h \times w \times 3}$ are two images with similar content then $||\varphi(x_1) - \varphi(x_2)||_{\ell_2}$ should be small). Feature extractors such as φ are an important component in many state-of-the-art methods in computer vision (see for example [13]).*

   To better understand the features extracted by φ, one might be motivated to analyze $D = \varphi(J)$. In this setting I is the list of all images in J, $Y = \mathbb{R}^r$, and f_D maps $i \in I$ to $\varphi(x_i)$. We analyze a specific example of this set-up in Section 3.1.

In order to put a topology on I, I must have some additional structure. Thus we assume that there exist subsets $U_1, U_2, \ldots, U_k$ of I that capture relationships between elements of I. This is the external information described in Section 1. We do not assume that $U_1, \ldots, U_k$ are disjoint.

Example 2. *We give some possible $U_1, \ldots, U_k$ for each part of Example 1.*

1. *(Toy example) Let $U_1 = \{a, b, c, d\}$ and $U_2 = \{c, d, e, f\}$ be known collaborations between researchers a, b, c, d, e, f.*

2. *(Sensors) For each $1 \leq p \leq k$ let U_p be the subset of I that contains all sensors of a certain type or measuring a given modality. Alternatively, let each U_p contain indices corresponding to sensors located in a given region.*

3. *(Gene expression) Subset U_p might contain all genes encoding proteins involved in a specific biochemical pathway p.*

4. *(Features extracted by a CNN) If the images in J are labeled based on whether they contain any of r different classes of object, then U_p might be the subset of I indexing all images with a specific object in it. For*

example, $x_i \in J$ might be an image containing a cat and a ball of yarn, in which case $i \in U_{cat}$ and $i \in U_{yarn}$.

Our proposed sheaf-theoretic framework requires three components: (i) a *topology* $\mathcal{T}_B$ on I built using $B = \{U_1, U_2, \ldots, U_k\}$, (ii) a *data sheaf* $\mathcal{D}$ on I where the raw data from datasets D is stored as a choice of sections, and (iii) a *model presheaf* $\mathcal{M}$ also on I which defines the type of model whose fit we want to understand.

Let $\mathcal{T}_B$ be the topology on I generated by subbasis $B = \{U_1, \ldots, U_k\}$. Note that $\mathcal{T}_B$ captures a notion of space on the elements of dataset D based on the external information found in B. In this paper $\mathcal{T}_B$ will underlie all the sheaves we construct unless otherwise specified. Since I is finite, then $\mathcal{T}_B$ is a finite topology.

In Section 3 we will compare sections of sheaves between open sets. In order to do this systematically we will take advantage of a lattice-theoretic viewpoint of $\mathcal{T}_B$. The open sets defining $\mathcal{T}_B$ have a poset structure, $(\mathcal{T}_B, \leq)$, based on set inclusion. That is, for $U, V \in \mathcal{T}_B$, $V \leq U \Leftrightarrow V \subseteq U$. In fact, $\mathcal{T}_B$ is a lattice since any two open sets U and V always have a meet and join given by their intersection and union respectively. For fixed open set U in $\mathcal{T}_B$, consider the order ideal generated from U, $\Lambda_U := \{V \in \mathcal{T}_B \mid V \subseteq U\}$. Since $\mathcal{T}_B$ is finite, and hence Λ_U is finite, there exists a finite filtration on Λ_U,

$$\{U\} = \Lambda_U^0 \subset \Lambda_U^1 \subset \cdots \subset \Lambda_U^k = \Lambda_U \tag{2}$$

where Λ_U^i consists of all those $V \in \Lambda_U$ such that there is a maximal chain in Λ_U, $V = V_j \subset V_{j-1} \subset \cdots \subset V_1 \subset U$ with $j \leq i$.

Example 3. *Consider the topology, depicted as a Haase diagram in Figure 1. Let $U = \{a, b, d\}$. Then*

$$\Lambda_U = \big\{U, \{a,d\}, \{a,b\}, \{a\}, \emptyset\big\} \tag{3}$$

with

$$\begin{aligned} \Lambda_U^0 &= \{U\}, \\ \Lambda_U^1 &= \big\{U, \{a,d\}, \{a,b\}\big\}, \\ \Lambda_U^2 &= \big\{U, \{a,d\}, \{a,b\}, \{a\}\big\}, \\ \Lambda_U^3 &= \big\{U, \{a,d\}, \{a,b\}, \{a\}, \emptyset\big\}. \end{aligned}$$

2.2. Data sheaves and model presheaves

Our goal in this section is to realize our dataset D as the section of a sheaf on the topological space $\mathcal{T}_B$ that we constructed in Section 2.1. We point the reader to [15, Part 1, Chapter 2] for an introduction to presheaves and sheaves.

Definition 2.1. *Let I be a finite set with topology generated by subbasis $B = \{U_1, \ldots, U_k\}$ and let Y be a set. Then the* data sheaf $\mathcal{D}$ *associated to (I, B, Y) is the sheaf that:*

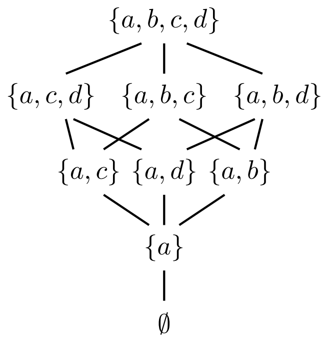

Figure 1. The Haase diagram of the finite topology from Example 3.

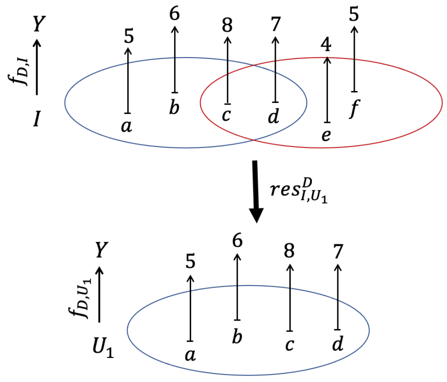

Figure 2. A visualization of the restriction map $\text{res}^{\mathcal{D}}_{I,U_1}$ that takes a section f_D from $\mathcal{D}(I)$ to f_{D,U_1} in $\mathcal{D}(U_1)$.

- *to each open set $U \subseteq I$ assigns the set $\mathcal{D}(U)$ of all functions from U to Y,*
- *and to each open set $V \subseteq U$ assigns the usual restriction of functions map $res^{\mathcal{D}}_{U,V} : \mathcal{D}(U) \to \mathcal{D}(V)$ which restricts functions from $U \to Y$ to functions from $V \to Y$.*

Remark 2.1. *Note that the fact that $\mathcal{D}$ is a sheaf, rather than just being a presheaf, follows from elementary arguments that spaces of functions (without further constraints) satisfy the gluing and locality axiom.*

Remark 2.2. *To make our constructions applicable to a large range of data science settings, we choose to work with sheaves taking values in the category* Set*. For specific problems, one can often choose to work with sheaves taking values in a category with more structure.*

Recall that we can realize D as a function $f_D : I \to Y$ which takes an element $i \in I$ as input and returns the measured value $f_D(i)$ at that point. The same construction holds for any subsets of I in the obvious way. Thus for any open set U of I, there is a map that takes a dataset D with index set I to a section of $\mathcal{D}(U)$, $f_{D,U} : U \to Y$. Together these define the map,

$$D \mapsto \{f_{D,U} : U \to Y \mid U \in \mathcal{T}_B\}. \tag{4}$$

From this perspective f_D is a global section of $\mathcal{D}$, in particular, $f_{D,I}$ in $\mathcal{D}(I)$.

Following [11], if $\mathcal{S}$ is a sheaf (or presheaf) on space X with topology $\mathcal{T}$, we call a choice of section a_U from each open set $U \in \mathcal{T}$,

$$A = \prod_{U \in \mathcal{T}} a_U, \tag{5}$$

an *assignment* on $\mathcal{S}$. We note that any global section a_X of $\mathcal{S}$ induces an assignment by setting $a_U = \text{res}_{X,U}(a_X)$ for each $U \in \mathcal{T}$.

D defines an assignment on $\mathcal{D}$ via global section f_D, $F_D = (f_{D,U})_{U \in \mathcal{T}_B}$.

Example 4. *(Toy example) There are* 5 *open subsets in the topology generated by subbasis* $U_1 = \{a, b, c, d\}$ *and* $U_2 = \{c, d, e, f\}$ *on* $I = \{a, b, c, d, e\}$*:* $\emptyset, \{c, d\}, U_1, U_2, I$*. The dataset D listed in Table 1, defines a function on each of these subsets. For example,* $f_{D,U_1} : U_1 \to \mathbb{R}$*, an element of* $\mathcal{D}(U_1)$*, is a function that sends* $f(a) = 5, \quad f(b) = 6, \quad f(c) = 8, \quad f(d) = 7$*. On the other hand,* $f_{D,\{c,d\}} : \{c, d\} \to \mathbb{R}$ *is the function that sends,* $f(c) = 8$ *and* $f(d) = 7$*. The full assignment* F_D *on* $\mathcal{D}$ *induced by D consists of* $f_{D,I}, f_{D,U_1}, f_{D,U_2}, f_{D,\{c,d\}}, f_{D,\emptyset}$*. Figure 2 contains a visualization of the restriction map* $res^{\mathcal{D}}_{I,U_1}$ *taking* $f_{D,I}$ *to* f_{D,U_1}*.*

Note that F_D contains a significant amount of redundant information, being completely determined by its single global section $f_D = f_{D,I}$. On the other hand, one can easily find assignments $A = (a_U)_{U \in \mathcal{T}_B}$ that are not determined by their associated global section a_I.

Example 5. *(Toy Example) We return to our publication example from Example 4. The reader can check that while the assignment described on the top in Table 2 is induced by a single global section* $f_{D,I} : I \to \mathbb{R}$ *which assigns*

$$f_I(a) = 4,\ f_I(b) = 4,\ f_I(c) = 2,\ f_I(d) = 3,$$
$$f_I(e) = 2,\ f_I(f) = 5,$$

the assignment provided on the bottom in Table 2 is not induced by a global section.

We give a special definition for those assignments that arise from a global section on a data sheaf (and hence from a dataset D).

Definition 2.2. *Let* $\mathcal{S}$ *be a sheaf (or presheaf) on space* X *with topology* $\mathcal{T}$*. An assignment* $A = (a_U)_{U \in \mathcal{T}}$ *is said to be* consistent *if for all* $U, V \in \mathcal{T}$ *with* $V \subseteq U$, $a_V = res_{U,V}(a_U)$*. Otherwise, A is said to be* inconsistent.

Section	a	b	c	d	e	f
f_I	4	4	2	3	2	5
f_{U_1}	4	4	2	3	✗	✗
f_{U_2}	✗	✗	2	3	2	5
$f_{\{c,d\}}$	✗	✗	2	3	✗	✗

Section	a	b	c	d	e	f
f_I	4	4	2	3	2	5
f_{U_1}	3	2	2	4	✗	✗
f_{U_2}	✗	✗	5	6	0	2
$f_{\{c,d\}}$	✗	✗	3	2	✗	✗

Table 2. Examples of assignments on the data sheaf from the toy publication example that do correspond to a global assignment (top) and do not correspond to a global assignment (bottom).

The assignment defined on top in Table 2 is consistent. The one on the bottom is inconsistent. It is clear that data assignment F_D induced by a dataset D is always consistent by construction.

2.3. The model presheaf and modeling map

In this section we define a presheaf on I. While the data sheaf $\mathcal{D}$ encodes instances of raw data collected on I, this new presheaf will encode local models of D. As in previous sections we assume that I is the space that indexes data from some dataset D and I has topology $\mathcal{T}_B$ generated by subbasis $B = \{U_1, \ldots, U_k\}$.

Definition 2.3. *Let* $\mathcal{D}$ *be a data sheaf on* I*. A* model presheaf *on* I *is a presheaf* $\mathcal{M}$ *on* I *along with a* modeling map $\Phi^{\mathcal{M}} = (\Phi^{\mathcal{M}}_U)_{U \in \mathcal{T}_B}$ *which consists of functions* $\Phi^{\mathcal{M}}_U : \mathcal{D}(U) \to \mathcal{M}(U)$ *for each* $U \in \mathcal{T}_B$*. For section* $f_U \in \mathcal{D}(U)$*, we call* $\Phi^{\mathcal{M}}_U(f_U)$ *a* model *of* f_U*.*

Remark 2.3. *Note that the modeling map* $\Phi^{\mathcal{M}}$ *is a critical component of the model presheaf* $\mathcal{M}$*. In fact without the modeling map,* $\mathcal{M}$ *is just a presheaf with no additional structure. Further,* $\Phi^{\mathcal{M}}$ *is generally not a presheaf morphism. In fact, we will derive a measure of global "model inconsistency" from the extent to which* $\Phi^{\mathcal{M}}$ *fails to be a presheaf morphism.*

Remark 2.4. *If we are handed a dataset, the data sheaf is generally implicit based on the form that the data takes. The space of sections associated with the model presheaf is chosen based on the type of models that one wants to use on the data. The least obvious choice in the framework that we present in this paper then is the definition of restriction maps in the model presheaf. In data science we rarely think* *about how to modify an existing model built to fit one dataset so that it instead fits a subset. When in doubt, we advocate keeping things simple. For open sets $V \subseteq U$, identity maps often work well provided that $\mathcal{M}(V) = \mathcal{M}(U)$.*

Example 6. *Below we give two examples of model presheaves.*

- *For each open set U in I, let $\mathcal{D}$ assign to U the space of functions from U to $\mathbb{R}$ (in other words, any dataset D defined by a global section of this data sheaf consists of real-valued measurements indexed by elements of I). Then we define the* averaging presheaf *$\mathcal{M}_{avg}$ to be the presheaf on I such that*

$$\mathcal{M}_{avg}(U) = \begin{cases} \{0\} & \text{if } U = \emptyset \\ \mathbb{R} & \text{otherwise}. \end{cases} \tag{6}$$

  with restriction maps given by

$$res^{\mathcal{M}_{avg}}_{U,V} = \begin{cases} 0 & \text{if } V = \emptyset \\ id & \text{otherwise} \end{cases} \tag{7}$$

  for $U, V \in \mathcal{T}_B$ with $V \subseteq U$. The model map $\Phi^{\mathcal{M}_{avg}}$ is defined such that for open set U in I,

$$\Phi^{\mathcal{M}_{avg}}_U(f_{D,U}) = \begin{cases} \frac{1}{|U|}\sum_{x\in U} f_{D,U}(x) & \text{if } U \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

  In other words, the average presheaf maps sections to their average. We could have chosen any number of different scalar valued statistics (for example, median, mode, maximum, etc.).

- *For each open set U in I, let $\mathcal{D}$ assign to U the space of functions from U to $\mathbb{R}^r$. For integer $q < r$, let $\mathcal{M}_{Graff(q,r)}$ be the constant presheaf such that*

$$\mathcal{M}_{Graff(q,r)}(U) = \begin{cases} \{0\} & \text{if } U = \emptyset \\ Graff(q,r) & \text{otherwise} \end{cases} \tag{9}$$

  where $Graff(q,r)$ is the Grassmannian of q-dimensional affine subspaces in $\mathbb{R}^r$ [7] (note that this is not to be confused with the affine Grassmannian that appears more commonly in algebraic geometry and representation theory). For simplicity we assume that restriction maps are given by:

$$res^{\mathcal{M}_{Graff(q,r)}}_{U,V} = \begin{cases} 0 & \text{if } V = \emptyset \\ id & \text{otherwise.} \end{cases} \tag{10}$$

  There are numerous approaches to finding a q-dimensional affine subspace W that best approximates vectors $v_1, \dots, v_m \in \mathbb{R}^r$ (assuming that $q < m$). Suppose that we have fixed such a method $G : \sqcup_{n\geq q} \underbrace{\mathbb{R}^r \times \cdots \times \mathbb{R}^r}_{n \text{ times}} \to Graff(q,r)$. Then we can define our model map $\Phi^{\mathcal{M}_{Graff(q,r)}}$ such that for $U = \{i_1, \dots, i_\ell\}$ and $f_{D,U}$ we set $\Phi^{\mathcal{M}_{Graff(q,r)}}_U(f_{D,U}) = G(f_{D,U}(i_1), \dots, f_{D,U}(i_\ell))$ if $U \neq \emptyset$ and is 0 otherwise. Note that for this construction to hold we need that if $U \neq \emptyset$ then $|U| \geq q$. In other words, q should be chosen based on the size of the smallest open set. One can also choose to vary q for different $U \in \mathcal{T}_B$, but in this case the identity can no longer be used for certain restriction maps.

Example 7. *(Toy example) Suppose that we construct an averaging presheaf based on researcher publication numbers from Example 1.1 Then to each open set (except $\emptyset$) in $I = \{a, b, c, d, e, f\}$ we associate $\mathbb{R}$. The model map $\Phi^{\mathcal{M}}$ then sends: $f_I \mapsto 5.8\overline{3}$, $f_{I,U_1} \mapsto 6.5$, $f_{I,U_2} \mapsto 6$, $f_{I,\{c,d\}} \mapsto 7.5$. We see that the two researchers c and d who are involved in multiple collaborations have more publications (on average) than those that do not. The difference between the average number of publications for c and d, $\Phi^{\mathcal{M}}(f_{I,\{c,d\}})$, and the average number of publications for all researchers in I is equal to*

$$|(res_{I,\{c,d\}} \circ \Phi^{\mathcal{M}})(f_I) - (\Phi^{\mathcal{M}} \circ res_{I,\{c,d\}})(f_I)|. \tag{11}$$

Note that this can also be interpreted as the extent to which $\Phi^{\mathcal{M}}$ fails to be a presheaf morphism with respect to restriction from the whole space I to open set $\{c, d\}$. We will explore this in more detail in Section 3.

3. Inconsistency of models

In Example 7 we saw that the inconsistency of the average number of publications across different subsets of collaborators could be identified with the extent to which restriction maps commute with the model map. This motivates the idea of model inconsistency. This notion was inspired by the idea of a consistency radius [11, Definition 20] in data fusion.

Definition 3.1. *Let*

- *D be a dataset with elements indexed by I and taking values in Y,*
- *$(I, \mathcal{T}_B)$ be the topological space associated with I and subbasis $B = \{U_1, \dots, U_k\}$,*
- *$\mathcal{D}$ be the data sheaf on I, $\mathcal{M}$ be a model presheaf on I, and $\Phi^{\mathcal{M}}$ be the model map taking spaces of sections from $\mathcal{D}$ to spaces of sections in $\mathcal{M}$,*
- *F_D be the assignment associated with D,*
- *and $d_U : \mathcal{M}(U) \times \mathcal{M}(U) \to \mathbb{R}_{\geq 0}$ be a metric for each $U \in \mathcal{T}_B$.*

Recall that for open set $U \subseteq I$, Λ_U is the order ideal defined by U (i.e. all open sets contained in U). The local inconsistency of assignment F_D with respect to $(\mathcal{D}, \mathcal{M})$ at U *is defined to be*

$$\mathit{Incon}(F_D, U) := \max_{V \in \Lambda_U} d_V(\mathit{res}^{\mathcal{M}}_{U,V}(\Phi^{\mathcal{M}}_U(f_{D,U})), \Phi^{\mathcal{M}}_V(f_{D,V})) \tag{12}$$

if Λ_U is non-empty and $\mathit{Incon}(F_D, U) = 0$ otherwise. The global inconsistency of F_D with respect to $(\mathcal{M}, \mathcal{D})$ *is defined to be*

$$\mathit{Incon}(F_D) := \max_{U \in \mathcal{T}_B} \mathit{Incon}(F_D, U). \tag{13}$$

Note that local inconsistency of assignment F_D with respect to the model presheaf/model map pair $(\mathcal{M}, \Phi^{\mathcal{M}})$ is closely tied to the extent to which the diagram

$$\begin{tikzcd} \mathcal{D}(U) \arrow[r, "\Phi^{\mathcal{M}}_U"] \arrow[d, "\mathrm{res}^{\mathcal{D}}_{U,V}"'] & \mathcal{M}(U) \arrow[d, "\mathrm{res}^{\mathcal{M}}_{U,V}"] \\ \mathcal{D}(V) \arrow[r, "\Phi^{\mathcal{M}}_V"] & \mathcal{M}(V) \end{tikzcd} \tag{14}$$

fails to commute. We formalize this in a proposition.

Proposition 3.1. *As above, let $(X, \mathcal{T}_B)$ be a topological space, $\mathcal{D}$ a data sheaf, $\mathcal{M}$ a model presheaf with model map $\Phi^{\mathcal{M}} : \mathcal{D} \to \mathcal{M}$, and $(d_U)_{U \in \mathcal{T}}$ be a collection of metrics. Then $\Phi^{\mathcal{M}}$ is a presheaf morphism if and only if for any consistent assignment $A = (f_U)_{U \in \mathcal{T}_B}$ of $\mathcal{D}$, the local inconsistency of A at any open set U is always 0.*

One direction of this proposition follows from the definition of a presheaf morphism (specifically the commutativity of restriction maps with each map $\Phi^{\mathcal{M}}_U$ between the spaces of sections). The other direction uses the fact that a section of a data sheaf can always be extended to a global section F on $\mathcal{D}(I)$.

Remark 3.2. *Note that we have chosen to define the local consistency of a model presheaf $\mathcal{M}$ with respect to section f_U by comparing $\mathit{res}^{\mathcal{M}}_{U,V}\Phi^{\mathcal{M}}_U(f_U)$ (the restriction of $\Phi^{\mathcal{M}}_U(f_U)$) against $\Phi^{\mathcal{M}}_V(f_V)$ for each V such that $V \subset U$. We could have conversely chosen to compare $\Phi^{\mathcal{M}}_U(f_U)$ against $\mathit{res}^{\mathcal{M}}_{W,U}\Phi^{\mathcal{M}}_W(f_W)$ for each W such that $U \subset W$. In the former case, if there are no proper, non-empty open sets V in U, then by definition the local inconsistency at U is 0 for all assignments. In the latter case the situation would be reversed and global sections would always have 0 inconsistency. We felt the idea of a model having lower inconsistency when fitted on smaller subsets aligns better with notions of fit from machine learning.*

Example 8. *(Toy example) We can compute the inconsistency of the consistent assignment in Table 1 with respect to the average model presheaf. We only need to define a distance function d_U on each copy of $\mathbb{R}$. We do this by choosing d_U to be the standard distance between real numbers: $d_U(x, y) = |x - y|$, for $x, y \in \mathbb{R}$. The local inconsistencies (omitting the empty set) are as follows: $\mathit{Incon}(A, I) = 3.67$, $\mathit{Incon}(A, U_1) = 1$, $\mathit{Incon}(A, U_2) = 1.5$, $\mathit{Incon}(A, \{c, d\}) = 0$. The global inconsistency is $\mathit{Incon}(A) = 3.67$. This occurs at the section on the total space I. This makes sense since we would expect a single statistic to represent a smaller set of numbers better than a larger set of numbers.*

Assume that $\mathcal{T}_B$ is a finite topology. Recall from Section 2.1 that for any $U \in \mathcal{T}_B$ we can form the order ideal Λ_U, which is filtered such that $\{U\} = \Lambda^0_U \subset \Lambda^1_U \subset \cdots \subset \Lambda^k_U = \Lambda_U$ for sufficiently large integer k. As illustrated in Example 8, it is often the case that the maximum value of

$$d_V(\mathrm{res}^{\mathcal{M}}_{U,V}(\Phi^{\mathcal{M}}_U(f_{D,U})), \Phi^{\mathcal{M}}_V(f_{D,V})) \tag{15}$$

from (12) is achieved for the smallest V contained in U, since in this case V and U have maximum difference (that is, $U \setminus V$ is maximally large). Informally, inconsistency is often maximized by restricting to the "smallest" open set V contained in U. In order to be able to "see past" this phenomenon, we introduce a final version of inconsistency which allows us to compare lack of commutativity of (14) only on "nearby" sets in the lattice associated with $\mathcal{T}_B$.

Definition 3.2. *Given the assumptions in Definition 3.1, if U is an open set in $\mathcal{T}_B$ and $\Lambda^1_U, \Lambda^2_U, \ldots, \Lambda^k_U$ is the $\mathbb{Z}$-filteration on the order ideal Λ_U, then the j*-filtered inconsistency *is defined as*

$$\mathit{Incon}_j(A, U) := \max_{V \in \Lambda^j_U} d_V(\mathit{res}^{\mathcal{M}}_{U,V}(\Phi^{\mathcal{M}}_U(f_U)), \Phi^{\mathcal{M}}_V(f_V)). \tag{16}$$

3.1. Example: Analyzing the feature space of a convolutional neural network

Many machine learning models built to perform classification tasks can be decomposed into two functions, (1) a *feature extractor* that takes a particular data type as input and extracts features most relevant to a task and (2) a simpler *prediction function* that uses these features to make predictions. A cat/dog classifier for example might consist of a convolutional neural network feature extractor, $\varphi : X \to \mathbb{R}^d$, and a predictive function, $\psi : \mathbb{R}^d \to \mathbb{R}^2$ (where one of the dimensions of $\mathbb{R}^2$ corresponds to the likelihood that the image is a dog and the other corresponds to the likelihood that the image is a cat). φ takes an image x from an image space X and encodes it as a vector in $\mathbb{R}^d$, translating the complex visual characteristics of cats and dogs into geometric structure in $\mathbb{R}^d$ that can then be separated by ψ.

When φ is a model with relatively few parameters and X is low-dimensional, there exist tools to probe the processes

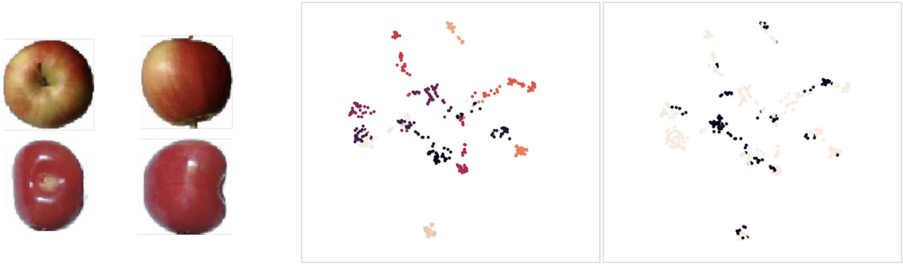

Figure 3. (Left) Example images from Fruits 360. (Center) A low-dimensional visualization of the image of the Fruits 360 dataset in the feature space of a convolutional neural network. Points are colored by type of fruit. (Right) The same visualization with points colored by whether the fruit stem (or stem node) is facing the camera or not.

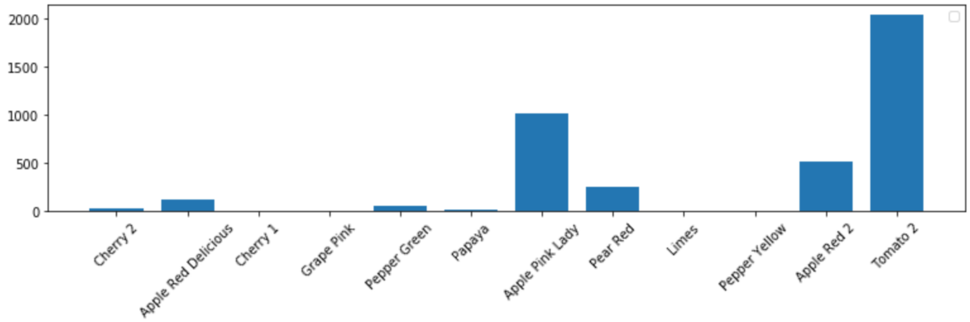

Figure 4. The number of times the removal of each type of fruit causes the maximum increase in "clusterability" with respect to (18).

by which the complete model $\psi \circ \varphi$ makes its decisions. However, in many state-of-the-art examples, φ is a large (for example over 20 million parameters), nonlinear model which has been trained on a large, high-dimensional dataset. Furthermore the feature space $\mathbb{R}^d$ of large feature extractors tend to themselves be high-dimensional, for example $d = 2048$ for the commonly used ResNet50 convolutional neural network [4]. Together all of these factors make current deep learning-based machine learning models difficult to analyze. At the same time, because these models are being used in a broader and broader range of applications the need for tools by which to analyze them has become increasingly pressing. In this section we give a very simple example of how the sheaf framework introduced in Sections 2.2, 2.3, and 3 can be used to answer questions about a deep learning-based feature extractor.

We take a very simple image dataset, *Fruits 360* [10], which consists of images of fruit photographed from various angles (see the left image in Figure 3 for examples). The standard task associated with this dataset is to predict what type of fruit is in each image, for example, *Apple*, *Cherry*, etc. If a feature extractor φ is large enough and has been exposed to enough image data it can easily extract the features necessary to differentiate between fruit even if φ has never been explicitly trained on this task. One observation made in [6] is that one can also naturally classify the images in this dataset based on whether or not the stem (or stem node) of the fruit is facing the camera or not. While humans can easily perform this task, the well-trained feature extractor φ mentioned above fails here. As described in [6], this is an example of a general trend wherein generic computer vision feature extractors tend to be very good at solving problems related to *class membership* (i.e. "does x belong to class y") but less good at problems involving quantity and orientation. The center and right images in Figure 3 give a 2-dimensional visualization of the images of the Fruits 360 dataset in $\mathbb{R}^{2048}$ under a feature extractor map φ trained on ImageNet. Comparison of the center image, which is colored by fruit type, and the right image, which is colored by whether the fruit's stem is facing the camera or not, suggests that φ strongly clusters by type of fruit while only clustering by stem/no-stem at the local level.

To better understand this limitation of φ we can ask the following question

Question: *Is φ poor at extracting the features required to solve the stem/no-stem problem for all the images in the Fruits 360 dataset or are there instead certain subpopulations on which φ particularly struggles dragging down the global performance?*

We can use our sheaf framework and filtered inconsistency to begin to answer this question with respect to type of fruit. Let D be the Fruits 360 dataset. We let I be an index of the images in D so that the dataset itself can be written as $\{x_i \mid i \in I\}$, where x_i is the image indexed by i. We let $B = \{U_1, \dots, U_k\}$ be defined such that, for example, U_j might contain all those elements of I that correspond to images containing the class *Apple*. The feature extractor we analyze is a ResNet50 convolutional neural network $\varphi : \mathbb{R}^{224\times 224\times 3} \to \mathbb{R}^{2048}$ that maps 224×224 RGB images to vectors in $\mathbb{R}^{2048}$. We load into φ model parameters from the Torchvision library [9] that were trained on ImageNet. Thus, φ has not been explicitly trained for either the standard Fruits 360 classification task on D nor the stem/no-stem task. For open set $U \in \mathcal{T}_B$, let the data sheaf $\mathcal{D}$ be defined such that $\mathcal{D}(U)$ consists of all functions from U to $\mathbb{R}^{2048}$. Then φ defines an assignment of $\mathcal{D}$ by defining the function $f_{D,U} : U \to \mathbb{R}^{2048}$ by $f_{D,U}(i) = \varphi(x_i)$.

Our model presheaf will be designed to capture the extent to which subsets of encoded images in $\mathbb{R}^{2048}$ tend to cluster based on their stem/no-stem labels. Inspired by [13], we measure the extent to which stem/no-stem images in open set U cluster, in the following way. Suppose $V_1 \subset U$ are those elements of U with the "stem" label and $V_2 \subset U$ consists of those elements of U with the "no stem" label, so that $V_1 \cup V_2 = U$. We randomly draw three examples i_1^s, i_2^s, i_3^s from V_1 and three examples $i_1^{ns}, i_2^{ns}, i_3^{ns}$ from V_1. Inspired by Prototypical Networks [13], a popular few-shot learning model, we use these to form *prototypes* for those elements of V_1 and V_2:

$$\gamma_s := \frac{1}{3}\sum_{j=1}^{3} f_{D,U}(i_j^s) \quad \text{and} \quad \gamma_{ns} := \frac{1}{3}\sum_{j=1}^{3} f_{D,U}(i_j^{ns}). \tag{17}$$

We predict the stem/no-stem label of the remaining elements $i \in U$ by solving the optimization problem:

$$\arg\min_{r=s,ns} ||\gamma_r - f_{D,U}(i)||. \tag{18}$$

We perform this process several times, recording our accuracy each time. We denote our average accuracy over many trials by $\alpha_{f,U}$. Note that $\alpha_{f,U}$ being closer to 1 indicates that $f_{D,U}$ more strongly clusters images based on the stem/no-stem property, since this means that more stem/no-stem labels can be predicted based on their proximity to prototypes for these classes γ_s and γ_{ns}.

Our model presheaf $\mathcal{M}$ is designed to store the values $\alpha_{f,U}$ and hence we assign to each $U \in \mathcal{T}_B$ the closed interval $[0,1]$. Our model map $\Phi^{\mathcal{M}} : \mathcal{D} \to \mathcal{M}$ is defined such that $f_{D,U} \in \mathcal{D}(U)$ is mapped to $\alpha_{f,U}$. Informally, $\Phi^{\mathcal{M}}$ sends the encoding of elements of U in $\mathbb{R}^{2048}$ to a score (based on (18)) that measures how well this encoding captures stem/no-stem clustering.

Because elements of the subbasis $B = \{U_1, \dots, U_k\}$ are mutually disjoint and their union is equal to I, $\mathcal{T}_B$ can be realized as the set of unions of all possible subsets of the sets in B, $\mathcal{T}_B = \{\cup_{W \in T} W \mid T \subseteq B\}$. Let $U \in \mathcal{T}_B$. Then $U = U_{j_1} \cup \cdots \cup U_{j_r}$ where $j_1, \dots, j_r \in \{1, \dots, k\}$. Then Λ_U^1 consists of sets V of the form $U_{j_1} \cup \cdots \cup \widehat{U_{j_t}} \cup \cdots \cup U_{j_r}$ where $1 \leq t \leq r$ and $\widehat{\ }$ denotes omission from the union. Thus for any open set U with $U \neq \emptyset$, Λ_U^1 consists of all those sets obtained by removing one type of fruit (that is, one U_j) from the union of subbasis elements that form U.

When calculating the local 1-filtered inconsistency for each $U \in \mathcal{T}_B$, we can not only record the 1-filtered inconsistency itself, but can note the $V \subset U$ at which the max (16) is achieved. Then $U \setminus V$ will correspond to a type of fruit. In Figure 4 we display a bar plot that shows the number of times each type of fruit appears when calculating this statistic. Rather than all types of fruit being equally problematic, we see that there are a few types that most frequently cause the largest drop in accuracy when they are included in the model's evaluation: *Tomato 2*, *Apple Pink Lady*, *Apple Red 2*, and *Pear Red*. It is not at all clear why the model tends not to cluster these fruits by their orientation. While *Tomato 2* does have a less visually distinctive stem node, *Apple Pink Lady* has not only a stem node but a stem itself which one would think a convolutional neural network would capture during feature extraction.

Note that our method is able to pick up on higher-order effects that result from the interactions of several specific fruit types. For example, perhaps the model is able to cluster stem/no-stem for *Apple Red 2* or *Pear Red* individually, but because of the way each is represented in $\mathbb{R}^{2048}$ the model struggles when they are together. For this reason, in the future it would be interesting to also examine Λ_U^2 and Λ_U^3.

4. Conclusion and Future Work

As statistical models grow ever bigger and more opaque, developing methods that give insight into not only the global behavior of the model, but also the local behavior becomes ever more important. In this work we develop a sheaf-theoretic framework to evaluate the fit of data-driven models. We show how a topology can be used to capture various subpopulations of interest within a dataset and then attach statistics related to model fit to each of these subpopulations. Via the notion of inconsistency, we can establish regions of the dataset on which a model's behavior changes significantly. We see application to real datasets as the next critical step in this direction. We expect that the tools developed here will need to be refined not only to meet real-world computational requirements, but also to highlight the aspects of model fit that the model builder actually cares about. Nevertheless, we hope that this work will increase the likelihood that sheaves will be a tool in some future model builder's toolbox.

References

[1] Justin Curry. Sheaves, cosheaves and applications. *arXiv preprint arXiv:1303.3255*, 2013.

[2] Jakob Hansen and Robert Ghrist. Learning sheaf laplacians from smooth signals. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5446–5450. IEEE, 2019.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] Cliff A Joslyn, Lauren Charles, Chris Depernoy, N Gould, K Nowak, B Praggastis, EA Purvine, M Robinson, J Strules, and P Whitney. A sheaf theoretical approach to uncertainty quantification of heterogeneous geolocation information. *Sensors*, 20:12:3418, 2020. https://doi.org/10.3390/s20123418.

[6] Henry Kvinge, Zachary New, Nico Courts, Jung H. Lee, Lauren A. Phillips, Courtney D. Corley, Aaron Tuor, Andrew Avila, and Nathan O. Hodas. Fuzzy simplicial networks: A topology-inspired model to improve task generalization in few-shot learning. In *AAAI Workshop on Meta-Learning and MetaDL Challenge*, volume 140 of *PMLR*, pages 77–89, 2021.

[7] Lek-Heng Lim, Ken Sze-Wai Wong, and Ke Ye. The Grassmannian of affine subspaces, 2020.

[8] Seyed MH Mansourbeigi. Sheaf theory approach to distributed applications: Analysing heterogeneous data in air traffic monitoring. *International Journal of Data Science and Analysis*, 3(5):34, 2017.

[9] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

[10] Horea Mureşan and Mihai Oltean. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 10(1):26–42, 2018.

[11] Michael Robinson. Sheaves are the canonical data structure for sensor integration. *Information Fusion*, 36:208–224, 2017.

[12] Olga Russakovsky, Jia Deng, Hao Su, Sanjeev Krause, Jonathan a nd Satheesh, Sean Ma, Zhiheng Huang, Aditya Karpathy, Andrej an d Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[13] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[14] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33, 2020.

[15] Ravi Vakil. The rising sea: Foundations of algebraic geometry notes, 2017.