Functorial Clustering via Simplicial Complexes

Dan Shiebler
University of Oxford
daniel.shiebler@kellogg.ox.ac.uk

Abstract

We adapt previous research on topological unsupervised learning to characterize a class of hierarchical overlapping clustering algorithms as functors that factor through a category of simplicial complexes. We first develop a pair of adjoint functors that map between simplicial complexes and the outputs of clustering algorithms. Next, we introduce the maximal and single linkage clustering algorithms as the respective composition of the flagification and connected components functors with a finite singular set functor. We then demonstrate that all other hierarchical overlapping clustering functors are refined by maximal linkage and refine single linkage.

1 Introduction

Say we have a finite dataset X that has been sampled from some larger space $\mathbf{X}$ according to a probability measure $\mu_{\mathbf{X}}$ over $\mathbf{X}$. Clustering algorithms group points in X together. In order for a clustering algorithm to be useful, the properties of its output must be somewhat in line with the structure of its input. One way to formalize this is to cast these algorithms as **functors**, or maps between categories that preserve identity morphisms and morphism composition. We can use this perspective to find commonalities between different algorithms, derive extensions of algorithms that preserve functoriality and identify modifications that break it. For example, Carlsson et al [CM13] use a functorial perspective to demonstrate that a broad class of non-overlapping clustering algorithms factor through single linkage and to develop a simple framework for generating clustering algorithms.

Our novel contributions include:

1. Definitions of maximal and single linkage in terms of the finite fuzzy singular set functor.
2. A reformulation of existing results on hierarchical overlapping clustering algorithms [CGHS16] in terms of functors that factor through a category of simplicial complexes.
3. A functorial strategy for using a finite clustering to partition an infinite space.

2 Uber-Metric Spaces, Simplicial Complexes, and Coverings

To stay consistent with McInnes et al [MHM18, McI19] and Spivak [Spi12], we represent datasets with **finite uber-metric spaces** (X, d_X). These metric-space-like objects permit $d(x_1, x_2)$ to be infinite and do not require $d(x_1, x_2) = 0$ to imply $x_1 = x_2$.

Definition 2.1. *In the category* $\mathbf{UMet}$ *objects are finite uber-metric spaces and morphisms are* ***non-expansive maps****, or functions* $f : X \to Y$ *such that* $d_Y(f(x_1), f(x_2)) \leq d_X(x_1, x_2)$.

Clustering algorithms group points in an uber-metric space together based on their distances. A useful tool for reasoning about groups of points in a finite set X is a finite simplicial complex:

Definition 2.2. *A* ***finite simplicial complex*** *is a family of finite sets that is closed under taking subsets.*

The finite sets in the simplicial complex are called **faces**, and the n**-simplices** are subcomplexes that contain all of the subsets of a single $(n + 1)$-element face in the complex. For example, the

Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

0-simplices (also called vertices) are points, the 1-simplices are pairs of points, the 2-simplices are triples of intersecting 1-simplices that we can visualize as triangles, etc. Note that any n-simplex in a simplicial complex is completely determined by the set of vertices that span it. We denote the set of n-simplices of the simplicial complex S_X as $S_X[n]$.

Definition 2.3. *The category* $\mathbf{SCpx}$ *has finite simplicial complexes as objects. The morphisms between the simplicial complex S_X with vertex set (0-simplices) X and S_Y with vertex set Y are **simplicial maps**, or functions $f : X \to Y$ such that if the vertices $\{x_1, x_2, ..., x_n\}$ span an n-simplex in S_X, then the vertices $\{f(x_1), f(x_2), ..., f(x_n)\}$ span an m-simplex in S_Y, where $m \leq n$.*

A particularly important class of simplicial complexes is the class determined by graphs:

Definition 2.4. *A **flag complex** is a simplicial complex which is generated by the cliques in a graph: the 0-simplices are the vertices, the 1-simplices are the pairs of vertices connected by an edge, and the n-simplices are the n-element cliques.*

The covers formed from the maximal simplices of flag complexes are particularly useful, since they are uniquely determined by the 2-element subsets of their elements [CGHS16]. In Section 3 we will define overlapping clustering algorithms as functors that map uber-metric spaces to these covers.

Definition 2.5. *Given a set X, a **non-nested flag cover** $\mathcal{C}_X$ of X is a cover of X such that: (1) if $A, B \in \mathcal{C}_X$ and $A \subseteq B$, then $A = B$, (2) the simplicial complex with vertices corresponding to the elements of X and faces all finite subsets of the sets in $\mathcal{C}_X$ is a flag complex.*

Definition 2.6. *The category* $\mathbf{Cov}$ *has tuples $(X, \mathcal{C}_X)$ as objects where $\mathcal{C}_X$ is a non-nested flag cover of the finite set X. The morphisms between $(X, \mathcal{C}_X)$ and $(Y, \mathcal{C}_Y)$ are functions $f : X \to Y$ where for any set S in $\mathcal{C}_X$ there exists some set S' in $\mathcal{C}_Y$ such that $f(S) \subseteq S'$.*

2.1 Flagification

Given a non-nested flag cover $(X, \mathcal{C}_X)$ in $\mathbf{Cov}$, we can generate a flag complex by treating the sets in $\mathcal{C}_X$ as the maximal simplices of the complex and adding in all of the subsets of these sets as the lower-dimensional simplices.

Definition 2.7. *The functor $S_{fl} : \mathbf{Cov} \to \mathbf{SCpx}$ maps the tuple $(X, \mathcal{C}_X)$ in $\mathbf{Cov}$ to the simplicial complex S_X whose n-simplices are the n-element subsets of sets in $\mathcal{C}_X$. Note that the set of vertices in S_X is X. S_{fl} acts as the identity on morphisms.*

Next, note that we can upgrade any cover U of a set X into a non-nested flag cover by iteratively adjoining to U any clusters mandated by the flag condition, and then removing all the non-maximal ones [CGHS16]. This procedure is called **flagification**. For example, consider some simplicial complex S_X with vertex set X. The simplices of S_X form a cover of X, which we can convert to a flag cover. We can use this procedure to develop the following functor:

Definition 2.8. *The functor $Flag : \mathbf{SCpx} \to \mathbf{Cov}$ maps the simplicial complex S_X to the tuple $(X, \mathcal{C}_X)$ where $\mathcal{C}_X$ is the flagification of the covering formed from the simplices of S_X. $Flag$ acts as the identity on morphisms.*

Proposition 1. *There exists an adjunction $S_{fl} \dashv Flag$. (Proof in 5.1)*

Another important functor is the connected components functor:

Definition 2.9. *The functor $\pi_0 : \mathbf{SCpx} \to \mathbf{Cov}$ has the same action on morphisms as $Flag$ but sends S_X to the tuple $(X, \mathcal{C}_X)$ where $\mathcal{C}_X$ is the set of connected components of S_X.*

2.2 Fuzzy Simplicial Complexes and Coverings

In order to examine how points in an uber-metric space group together at different scales, we switch to fuzzy simplicial complexes, which are a simplification of Spivak's fuzzy simplicial sets [Spi12]. In a fuzzy simplicial complex, each simplex is present with a certain strength. We represent simplex strengths with elements in the partially ordered set $I = (0, 1]$ under the relation $\leq$.

Definition 2.10. *A **fibered fuzzy simplicial complex** is a functor $F_X : I^{op} \to \mathbf{SCpx}$ such that for any morphism $a \leq a'$ in I^{op}, the simplicial map $F_X(a \leq a')$ acts as the identity on 0-simplices.*

In a fibered fuzzy simplicial complex F_X the vertex set of the simplicial complex $F_X(a)$ is the same for all a in $(0, 1]$. For simplicity we will also call this set the **vertex set** of F_X. The functoriality of F_X then implies that for $a, a' \in (0, 1]$ such that $a \leq a'$, if the n-simplex σ is in $F_X(a')$ then σ is also in $F_X(a)$. The **strength** of σ in the fibered fuzzy simplicial complex F_X is the largest a in $(0, 1]$

such that σ is in $F_X(a)$. Note also that for any $S \subseteq (0, 1]$ and fibered fuzzy simplicial complex F_X, the collection $\{F_X(a) \mid a \in S\}$ is a filtration of simplicial complexes [CM17].

Definition 2.11. *The objects in the category* $\mathbf{FSCpx}$ *are fibered fuzzy simplicial complexes and the morphisms are natural transformations.*

We use a similar construction to build a category of fuzzy non-nested covers, which are similar to Culbertson et al's persistent covers [CGHS16, CGS18]:

Definition 2.12. *The objects in the category* $\mathbf{FCov}$ *are functors* $F_X : I^{op} \to \mathbf{Cov}$ *such that* $S_{fl} \circ F_X$ *is a fibered fuzzy simplicial complex. The morphisms in* $\mathbf{FCov}$ *are natural transformations.*

We can now define the functors $(S_{fl} \circ -) : \mathbf{FCov} \to \mathbf{FSCpx}$ and $(Flag \circ -) : \mathbf{FSCpx} \to \mathbf{FCov}$ in order to build the functor $FlagCpx = (S_{fl} \circ -) \circ (Flag \circ -)$, which converts any fuzzy simplicial complex into a fuzzy simplicial flag complex.

2.3 The Realization and Singular Set Functors

Spivak [Spi12] and McInnes et al [MHM18] describe the adjoint functors $FinReal$ and $FinSing$ which act similarly to the Realization and Singular Set functors from algebraic topology and transform between fuzzy simplicial sets and uber-metric spaces. In this subsection we slightly adapt these functors to transform between fibered fuzzy simplicial complexes and uber-metric spaces.

Definition 2.13. *The functor* $Pair : \mathbf{UMet} \to \mathbf{FSCpx}$ *sends the uber-metric space* $(X, d_X) \in \mathbf{UMet}$ *to the fibered fuzzy simplicial complex* $F_X : I^{op} \to \mathbf{FSCpx}$ *where for* $a \in (0, 1]$*,* $F_X(a)$ *is a simplicial complex whose set of* 0*-simplices is* X *and whose* 1*-simplices are the pairs* $\{x_1, x_2\} \subseteq X$ *such that* $d_X(x_1, x_2) \leq -\log(a)$*.* $F_X(a)$ *has no* n*-simplices for* $n > 1$*.* $Pair$ *maps the morphism* $f : X \to Y$ *to the natural transformation whose component at* a *is* f*.*

Proposition 2. $Pair$ *is a functor. (Proof in 5.2)*

We use $-\log$ because it is a monotonically decreasing function from $(0, 1]$ to $[0, \infty)$. That is, if $d_X(x_1, x_2) = 0$, then the strength of the simplex $\{x_1, x_2\}$ in the fibered fuzzy simplicial complex $Pair\ (X, d_X)$ is 1, whereas when $d_X(x_1, x_2)$ approaches ∞, the strength of the simplex $\{x_1, x_2\}$ approaches 0. We can now define the functor $FinSing : \mathbf{UMet} \to \mathbf{FSCpx}$ as:

$$FinSing = FlagCpx \circ Pair$$

For $a \in (0, 1]$, $FinSing(X, d_X)(a)$ is a flag complex in which the set $\{x_1, x_2, ..., x_n\} \subseteq X$ forms a simplex if all pairwise distances between points in the set are less than $-\log(a)$. This is the **Vietoris-Rips Complex** of (X, d_X) at $-\log(a)$.

Definition 2.14. *The functor* $FinReal : \mathbf{FSCpx} \to \mathbf{UMet}$ *maps* $F_X : I^{op} \to \mathbf{SCpx}$ *with vertex set* X *to* (X, d_X) *such that* $d_X(x_1, x_2) = inf\{-\log(a) \mid a \in (0, 1], \{x_1, x_2\} \in F_X(a)[1]\}$*. On morphisms,* $FinReal$ *sends a natural transformation* $\mu : F_X \to F_Y$ *to the function determined by* μ*'s actions on the vertex set of* F_X*. Note that this function must be non-expansive since for all* $a \in (0, 1]$*, if* $\{x_1, x_2\} \in F_X(a)[1]$ *then* $\{f(x_1), f(x_2)\}$ *is in* $F_Y(a)[0]$ *or* $F_Y(a)[1]$*.*

Intuitively, $FinReal\ F_X$ is an uber-metric space where the distance between the points x_1, x_2 is determined by the strength of the 1-simplex connecting them (if any).

Proposition 3. *There exists an adjunction* $FinReal \dashv FinSing$*. (Proof in 5.3)*

3 Clustering With Overlaps

Overlapping clustering algorithms accept finite uber-metric spaces (X, d_X) and return non-nested flag covers of X.

Definition 3.1. *A* ***flat clustering functor*** *is a functor* $C : \mathbf{UMet} \to \mathbf{Cov}$ *that is the identity on the underlying set.*

Now define Λ^1_ϵ to be the two-element metric space where the distance between the two elements is $\epsilon \in \mathbb{R}_{\geq 0}$. Most useful flat clustering functors are in the following class.

Definition 3.2. *A* ***non-trivial flat clustering functor*** C *is a flat clustering functor such that there exists a* ***clustering parameter*** $\delta_C \in \mathbb{R}_{\geq 0}$ *where* $C(\Lambda^1_\epsilon)$ *is two singleton clusters for any* $\epsilon > \delta_C$ *and a single two point cluster for any* $\epsilon \leq \delta_C$

Intuitively, the clustering parameter δ_C is the maximum distance at which the clustering functor will identify two points as being part of the same cluster.

3.1 Extrapolating a Clustering

In practice we often need to extrapolate a clustering to out-of-sample points. Say we have a flat clustering functor C, a not-necessarily-finite uber-metric space $(\mathbf{X}, d_{\mathbf{X}})$, and some finite $X \subset \mathbf{X}$. We want to produce a covering of $(\mathbf{X}, d_{\mathbf{X}})$ by grouping the points in $\mathbf{X} - X$ into the sets in $C(X, d_{\mathbf{X}})$.

To do this, first define a functor $C_{X\cup\{x\}}$ that maps uber-metric spaces of the form $(X \cup \{x\}, d_{\mathbf{X}})$ to the maximal cover that is refined by $C(X, d_{\mathbf{X}}) \cup \{\{x\}\}$ and refines $C(X \cup \{x\}, d_{\mathbf{X}})$. The cover $C_{X\cup\{x\}}(X \cup \{x\}, d_{\mathbf{X}})$ is identical to $C(X, d_{\mathbf{X}})$, except some of the sets in this cover will also contain the point x. Intuitively, $C_{X\cup\{x\}}$ assigns each $x \in \mathbf{X} - X$ to the sets in $C(X, d_{\mathbf{X}})$ that contain the points in X that share a cluster with x in $C(X \cup \{x\}, d_{\mathbf{X}})$. In order to stitch together each of these assignments into a cover of $\mathbf{X}$, we can simply take the colimit of the functor $C_{X\cup\{x\}}$. Intuitively, this colimit is a cover of $\mathbf{X}$ that is refined by $C(X, d_{\mathbf{X}}) \cup \{\{x_i\} \mid x_i \in \mathbf{X} - X\}$.

3.2 Hierarchical Clustering Functor

One of the largest challenges when building clustering algorithms is the difficulty in capturing structure that exists at different scales. The **Kleinberg Impossibility Theorem** states that any scale invariant clustering algorithm must sacrifice either surjectivity or consistency [Kle03]. One way to get around this is to generate a series of clusterings, each at a different scale:

Definition 3.3. *A **hierarchical clustering functor** is a functor $H : \mathbf{UMet} \to \mathbf{FCov}$ such that for $a \in (0, 1]$, $H(-)(a) : \mathbf{UMet} \to \mathbf{Cov}$ is a flat clustering functor.*

Definition 3.4. *A **non-trivial hierarchical clustering functor** H is a hierarchical clustering functor such that for all $a \in (0, 1]$, $H(-)(a) : \mathbf{UMet} \to \mathbf{Cov}$ is a non-trivial flat clustering functor with clustering parameter $\delta_{H,a}$.*

Hierarchical clustering functors are similar to Culbertson et al's sieving functors [CGHS16].

3.3 Maximum Linkage and Single Linkage

The **single linkage** algorithm $\mathcal{SL}$ generates a covering of X such that the points $x_1, x_2 \in X$ lie in the same cluster with strength at least a if there exists a sequence of points $x_1, x_i, x_{i+1}, ..., x_{i+n}, x_2$ such that the largest distance between any adjacent points in the sequence is no larger than $-\log(a)$. In contrast, the **maximal linkage** algorithm $\mathcal{ML}$ generates a covering of X such that the points $x_1, x_2 \in X$ lie in the same cluster with strength at least a if the distance between them is no larger than $-\log(a)$. Given a uber-metric space (X, d_X), we can generate the maximal linkage and single linkage clusterings by first forming the Vietoris-Rips complex with the $FinSing$ functor and then taking the maximal simplices and the connected components of this complex respectively.

$$\mathcal{ML} = (Flag \circ -) \circ FinSing \qquad \mathcal{SL} = (\pi_0 \circ -) \circ FinSing$$

Note that the functors $(Flag \circ -)$ and $(\pi_0 \circ -)$ respectively map the functor $F_X \in \mathbf{FSCpx}$ to $Flag \circ F_X$ and $\pi_0 \circ F_X$.

Proposition 4. *$\mathcal{ML}$ and $\mathcal{SL}$ are non-trivial hierarchical clustering functors. (Proof in 5.4)*

Single linkage always generates a partition of X, whereas maximal linkage does not. Furthermore, unlike with single linkage it is possible to reconstruct an uber-metric space from its hierarchical maximal linkage clustering:

Proposition 5. *There exists an adjunction $FinReal \circ (S_{fl} \circ -) \dashv \mathcal{ML}$. (Proof in 5.5)*

Intuitively, single linkage and maximal linkage clustering lie on two ends of a spectrum of clustering refinement. Any other non-trivial hierarchical clustering functor lies between them. Formally, we can make the following claim, which is inspired by Theorem 8 in Culbertson et al [CGHS16]:

Proposition 6. *Suppose $H : \mathbf{UMet} \to \mathbf{FCov}$ is a non-trivial hierarchical clustering functor such that for all $a \in (0, 1]$, the functor $H(-)(a) : \mathbf{UMet} \to \mathbf{Cov}$ has clustering parameter $\delta_{H,a}$ and define $W_H(a) = e^{-\delta_{H,a}}$. Then there exist natural transformations with inclusion maps as components from $\mathcal{ML}(-)(W_H(-))$ to H and from H to $\mathcal{SL}(-)(W_H(-))$. (Proof in 5.6)*

4 Conclusion

In this paper we introduce a framework for describing clustering algorithms as functors between categories of uber-metric spaces, simplicial complexes, and coverings. This strategy allows us to directly model the invariants an algorithm preserves, reason about similarities and hierarchies between algorithms or characterize them based on their functorial properties.

References

[CGHS16] Jared Culbertson, Dan P Guralnik, Jakob Hansen, and Peter F Stiller. Consistency constraints for overlapping data clustering. *arXiv preprint arXiv:1608.04331*, 2016.

[CGS18] Jared Culbertson, Dan P Guralnik, and Peter F Stiller. Functorial hierarchical clustering with overlaps. *Discrete Applied Mathematics*, 236:108–123, 2018.

[CM13] Gunnar Carlsson and Facundo Mémoli. Classifying clustering schemes. *Foundations of Computational Mathematics*, 13(2):221–252, 2013.

[CM17] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.

[Kle03] Jon M Kleinberg. An impossibility theorem for clustering. In *Advances in Neural Information Processing Systems*, pages 463–470, 2003.

[McI19] Leland McInnes. Topological methods for unsupervised learning. In *International Conference on Geometric Science of Information*, pages 343–350. Springer, 2019.

[MHM18] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[Spi12] David I Spivak. Metric realization of fuzzy simplicial sets. *Self published notes*, 2012.

5 Appendix

5.1 Proof of Proposition 1

Proof. We will show that there exists an isomorphism:

$$\mathbf{SCpx}[S_{fl}\ c, s] \simeq \mathbf{Cov}[c, Flag\ s]$$

That is natural over $s \in ob(\mathbf{SCpx})$ and $c \in ob(\mathbf{Cov})$.

Consider first the function $f_{c,s} : \mathbf{SCpx}[S_{fl}\ c, s] \to \mathbf{Cov}[c, Flag\ s]$ that sends a simplicial map $h : S_{fl}\ c \to s$ to the function defined by its action on vertices. We will show that this function is a morphism in $\mathbf{Cov}[c, Flag\ s]$. Consider the set $S \in c$. For any $x_1, x_2 \in S$ the simplex $\{x_1, x_2\}$ must be in $S_{fl}\ c$. Since h is a simplicial map this implies that the simplex $\{h(x_1), h(x_2)\}$ is in s, which implies that there must exist some set in $Flag\ s$ that contains $\{x_1, x_2\}$ and therefore S.

Next, consider the function $g_{c,s} : \mathbf{Cov}[c, Flag\ s] \to \mathbf{SCpx}[S_{fl}\ c, s]$ that sends a function $h : c \to Flag\ s$ to the map that applies h to the vertices of $S_{fl}\ c$. We will show that this is a simplicial map in $\mathbf{SCpx}[S_{fl}\ c, s]$. Note that since $S_{fl}\ c$ is a flag complex we only need to show that this simplicial map preserves 1-simplices. Consider the 1-simplex $\{x_1, x_2\} \in S_{fl}\ c$. There must exist some set S in c such that $\{x_1, x_2\} \subseteq S$, which implies that there exists some set S' in $Flag\ s$ such that $\{h(x_1), h(x_2)\} \subseteq S$. Then we can conclude that the simplex $\{h(x_1), h(x_2)\} \in s$.

Since these maps do not modify the action of a function over vertices it is clear that they are inverses and are both natural in c and s. □

5.2 Proof of Proposition 2

Proof. $Pair$ trivially preserves identity and composition, so we just need to show that for $a \in (0, 1]$, the function h is a simplicial map from $Pair(X, d_X)(a)$ to $Pair(Y, d_Y)(a)$. Since $Pair(X, d_X)(a)$ is a flag complex, we simply need to demonstrate that h preserves 1-simplices. Now suppose $\{x_1, x_2\}$ is a simplex in $Pair(X, d_X)(a)$. Then by the definition of $Pair$ we have $d_X(x_1, x_2) \leq -\log(a)$, which by the non-expansiveness of h implies that $d_Y(h(x_1), h(x_2)) \leq -\log(a)$, so $\{h(x_1), h(x_2)\}$ is a simplex in $Pair(Y, d_Y)(a)$ □

5.3 Proof of Proposition 3

Proof. We will show that there exists an isomorphism:

$$\mathbf{UMet}[FinReal\ c, m] \simeq \mathbf{FSCpx}[c, FinSing\ m]$$

That is natural over $m \in ob(\mathbf{UMet})$ and $c \in ob(\mathbf{FSCpx})$.

Consider first the function $f_{c,m} : \mathbf{UMet}[FinReal\ c, m] \to \mathbf{FSCpx}[c, FinSing\ m]$ that sends a non-expansive map $h : FinReal\ c \to m$ to the simplicial map that applies h to the vertices of c. We will show that this function is a simplicial map in $\mathbf{FSCpx}[c, FinSing\ m]$. Note that since $FinSing\ m$ is a flag complex we only need to show that this simplicial map preserves 1-simplices. For any $a \in (0, 1]$ suppose there exists a 1-simplex $\{x_1, x_2\}$ in $c(a)$. Then the distance between x_1 and x_2 in $FinReal\ c$ must be less than $-\log(a)$, which implies that the distance between $h(x_1)$ and $h(x_2)$ in m must be less than $-\log(a)$ as well. This implies that $\{h(x_1), h(x_2)\}$ spans a simplex in $(FinSing\ m)(a)$

Next, consider the function $g_{c,m} : \mathbf{FSCpx}[c, FinSing\ m] \to \mathbf{UMet}[FinReal\ c, m]$ that sends a simplicial map $h : c \to FinSing\ m$ to the function that is defined by the action of h on the vertices of $FinReal\ c$. We will show that this is a non-expansive map in $\mathbf{UMet}[FinReal\ c, m]$. Suppose that for some $a \in (0, 1]$ the distance between the points x_1, x_2 in $FinReal\ c$ is less than $-\log(a)$. Then the simplex $\{x_1, x_2\}$ exists in c with strength at least a, so the simplex $\{h(x_1), h(x_2)\}$ exists in $FinSing\ m$ with strength at least a. This implies that the distance between $h(x_1), h(x_2)$ in m is less than $-\log(a)$.

It is clear that both of these functions are natural in c and s and that they are inverses. □

5.4 Proof of Proposition 4

Proof. First, we note that $\mathcal{ML}$ and $\mathcal{SL}$ are hierarchical clustering functors. Given an uber-metric space (X, d_X), the vertex set of $FinSing\ (X, d_X)$ is X. Since both $(Flag \circ -)$ and $(\pi_0 \circ -)$ map a fuzzy simplicial complex with vertex set X to a fuzzy covering of X, $\mathcal{ML}(-)(a)$ and $\mathcal{SL}(-)(a)$ commute with the forgetful functor.

Next, $\mathcal{ML}$ and $\mathcal{SL}$ are non-trivial since for $a \in (0, 1]$ the functors $\mathcal{ML}(-)(a)$ and $\mathcal{SL}(-)(a)$ have the clustering parameter $-\log(a)$. □

5.5 Proof of Proposition 5

Proof. Since $\mathcal{ML} = (Flag \circ -) \circ FinSing$ we can express this as

$$FinReal \circ (S_{fl} \circ -) \dashv (Flag \circ -) \circ FinSing$$

which holds by the composition of adjunctions since $FinReal \dashv FinSing$ by Proposition 3 and $(S_{fl} \circ -) \dashv (Flag \circ -)$ by Proposition 1. □

5.6 Proof of Proposition 6

Proof. First, we construct a natural transformation from $\mathcal{ML}(-)(W_H(-))$ to H by demonstrating that for all $(X, d_X) \in \mathbf{UMet}, a \in (0, 1]$, the identity function on X is a morphism in $\mathbf{Cov}$ from $\mathcal{ML}(X, d_X)(W_H(a))$ to $H(X, d_X)(a)$. Consider the set $\{x_1, x_2, ..., x_n\} \in \mathcal{ML}(X, d_X)(W_H(a))$. The maximum distance between any two points in this set must less than $\delta_{H,a}$. Therefore, for any x_i, x_j in this collection, there exists a morphism in $\mathbf{UMet}$ from $\Lambda^1_{\delta_{H,a}}$ to (X, d_X) that sends the two elements of $\Lambda^1_{\delta_{H,a}}$ to x_i and x_j respectively. Since H is a functor and $H(\Lambda^1_{\delta_{H,a}})$ is a single two-point cluster, this implies that there is a set in the cover $H(X, d_X)(a)$ that contains both x_i and x_j. Since $H(X, d_X)(a)$ is flag, there must exist a set $S \in H(X, d_X)(a)$ such that $\{x_1, x_2, ..., x_n\} \subseteq S$, which implies that the identity function on X is a morphism in $\mathbf{Cov}$ from $\mathcal{ML}(X, d_X)(W_H(a))$ to $H(X, d_X)(a)$.

Next, we construct a natural transformation from H to $\mathcal{SL}(-)(W_H(-))$ by demonstrating that for all $a \in (0, 1]$, the identity function on X is a morphism in $\mathbf{Cov}$ from $H(X, d_X)(a)$ to $\mathcal{SL}(X, d_X)(W_H(a))$. Suppose there exist $x_1, x_2 \in X$ such that there is no $S \in \mathcal{SL}(X, d_X)(W_H(a))$ where $\{x_1, x_2\} \subseteq S$. Then for some $\epsilon > \delta_{H,a}$ there exists a morphism in $\mathbf{UMet}$ that sends all x_1' that share a set in $\mathcal{SL}(X, d_X)(W_H(a))$ with x_1 and all x_2' that share a set in $\mathcal{SL}(X, d_X)(W_H(a))$

with x_2 to different points. Since H is a functor this implies that there does not exist any S in $H(X, d_X)(a)$ such that $\{x_1, x_2\} \subseteq S$. We can therefore conclude that the identity function on X is a morphism in **Cov** from $H(X, d_X)(a)$ to $\mathcal{SL}(X, d_X)(W_H(a))$. $\square$