HIGHER-DIMENSIONAL MODELS OF SOCIAL NETWORKS

DAVID I. SPIVAK

Abstract. Graphs are often used to model social networks: the nodes represent agents and the edges represent lines of communication. Often one wishes to model a situation in which communication can occur between groups of more than two people, and in these situations graphs are insufficient as models. In these cases, the researcher may use hypergraphs whose hyperedges represent these larger groups.

In this paper we give a well-defined category of models for higher-dimensional networks, of which hypergraphs is a main example. Another primary example is that of simplicial sets, which can provide a geometric intuition for higher-dimensional graph theory. Simplicial sets have been a favorite tool in the mathematical field of algebraic topology for over 70 years. We show that simplicial sets succeed as models for social networks in many cases that hypergraphs do not.

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Mathematical definitions</td>
<td>3</td>
</tr>
<tr>
<td>3. Choosing the appropriate model</td>
<td>11</td>
</tr>
<tr>
<td>4. Applications and further research</td>
<td>15</td>
</tr>
</tbody>
</table>

1. Introduction

Q-Analysis!!!!!!

1) Linking event.

2) Cliques

Networks are commonly conceived as directed or undirected graphs ([?], [?],[?]), in which the nodes represent entities and the edges represent lines of communication. This conception works well in many circumstances, but when its purpose is to model how communication works in the real world, it suffers from a glaring shortcoming: communication can take place in groups of more than two entities.

Let us imagine a table of four people having a conversation together at a restaurant; this event is understood as a social network with four nodes. If everyone can hear everyone else, then graph-theoretically this network is represented by a complete graph on four vertices. Now imagine a situation of four people playing the game “telephone” so that each person may only whisper in the ear of another person. If each person can whisper to anyone else, this scenario is again modeled graph-theoretically by a complete graph on four vertices. But the situations are

This project was supported in part by the Office of Naval Research.
extremely different! One-dimensional graph theory does not capture the distinction between a single 4-person conversation and six 2-person conversations.

In this paper, we will provide higher-dimensional models that easily capture the distinction. Hypergraphs form a category of such models (see [?] and have been studied at least since the 1960s (by Erdős, for example, see [?]). Just as a graph consists of a set of vertices and a set of edges, each of which connects two vertices, a hypergraph consists of a set of vertices and a set of hyperedges, each of which connects some finite number of vertices. Such hyperedges can be used to encode events in which communication takes place among more than two people.

However, hypergraphs are only one solution to the problem, and they are inadequate for many purposes. The three main drawbacks of hypergraphs are: they are difficult to visualize; they are difficult to analyze topologically; and they do not allow one to encode sub-contexts, in the sense that one cannot posit any relationship between two hyperedges. We make this last problem explicit below and call it the sub-edge problem; see Remark 3.5.

In contrast, simplicial sets offer a much richer category of models for higher dimensional graphs (see Section 2.20 for the comparison). One draws a hypergraph by drawing a set of dots and enclosing certain subsets of them within circles. Simplicial sets, on the other hand, are visualized as spaces, more specifically as multi-dimensional polygonal shapes made up of vertices, edges, triangles, and higher-dimensional triangles like tetrahedra. An $n$-dimensional triangle (called an $n$-simplex) can be thought of as the “polyhedral hull” of $n + 1$ vertices. Thus, a 1-dimensional triangle, or 1-simplex, is the polyhedral hull of 2 vertices: it is simply an edge. Likewise, a 2-simplex is the hull of 3 vertices and is hence a triangle; a 3-simplex is a tetrahedron; and a 0-simplex is just a vertex.

In general, a simplicial set is the union of many such vertices, edges, triangles, etc. To make this a little more clear, let us emphasize that any graph is a 1-dimensional simplicial set. Now suppose we want to make a 2-dimensional simplicial set. Begin with a graph $G$, and choose a set of three edges $(a,b)$, $(b,c)$, and $(a,c)$ which form a triangle inside $G$:

Given such a triangle, one may attach in a 2-simplex to $G$, filling in the triangle $abc$. Thus a 2-dimensional simplicial set looks like a graph except that some triangles are filled. To create a 3-dimensional simplicial set, begin with a 2-dimensional simplicial set, choose some groupings of four triangles that form an empty tetrahedron, and fill them. At this point, the reader can imagine constructing arbitrary simplicial sets. An $n$-dimensional simplicial set is one in which the largest simplex is an $n$-simplex.

The appropriate simplicial set model for the group of four people communicating together is the tetrahedron. This solid shape represents the idea that the conversation is taking place between four entities in a shared space. The tetrahedron has six edges that together make up the 1-dimensional “skeleton” of the tetrahedron. One checks that the graph comprised of these six edges is the complete graph on four vertices. This is the appropriate simplicial set model for the situation in which the four people may only whisper to each other.
As mentioned above, there is a single framework that includes both hypergraphs and simplicial sets, as well as countless other examples, many of which have interesting uses as models for certain types of networks. We make this framework explicit in Section 2 where we also give mathematical definitions for all of the higher-dimensional models we discuss in this paper. In Section 3 we give several examples of real-world networks (including those of computer processes, military missions, and conference calls) and explain how one chooses the mathematical model which best describes each one. Finally, in Section 4 we discuss further applications of these ideas as well as directions for future research.

1.1. Related Research. Before leaving the introduction, we discuss some related work. Simplicial sets have been used before in computer science. In [?], the authors use simplicial sets to model word associations in documents, so that highly connected components correspond to primitive concepts. In [?], the author uses simplicial sets to model databases. Here the simplices represent tables, and overlapping simplices represent tables that share a common set of attributes. Finally many researchers ([?],[?]) are using simplicial sets to model large data sets. Given a data set, like a jpeg file or an MRI scan, they connect sets of \( n+1 \) vertices with an \( n \)-simplex whenever these vertices are sufficiently close together. They then use an invariant called homology to analyze the shape. For example, the authors in [?] have found that in the brain’s visual cortex, lines are represented by a Klein Bottle.

1.2. Acknowledgments. The author would like to thank Paea LePendu for many useful conversations.

2. Mathematical definitions

In this section we give rigorous mathematical definitions for graphs, hypergraphs, simplicial sets, etc. We begin with background material (Section 2.1), such as the definition of graphs and hypergraphs, and their respective morphisms. We then proceed to give a set-theoretic definition of semi-simplicial sets, which are a slightly less sophisticated version of simplicial sets, in Section 2.7.

In Section 2.12 we switch gears and begin to view everything from a category-theoretic standpoint. We redefine graphs, hypergraphs, and semi-simplicial sets as categories of functors, and show that these definitions match with the set-theoretic ones given above. We explain how to compare these various models in Section 2.20 and then finally generalize everything in this section by defining categories of higher graphs and giving some new examples in Section 2.22.

2.1. Background. A graph consists of a set \( E \) of edges, a set \( V \) of vertices, and a way to know which two vertices belong to a given edge. We allow multiple edges to connect the same two vertices, and we allow loops. Precisely, we have the following definition.

**Definition 2.2.** A directed graph is a triple \( X = (X_E, X_V, f) \), where \( X_E \) and \( X_V \) are sets (called the set of edges and the set of vertices respectively), and where \( f : X_E \to X_V \times X_V \) is a function, which we call the constituent function.

A hypergraph is like a graph in that it has vertices and (hyper-) edges, but the set of vertices contained in a hyperedge can have cardinality other than 2. For a set
$S$, let $\mathbb{P}_+(S)$ denote the set $\{A \subseteq S | A \neq \emptyset\}$ of non-empty subsets of $S$. Recall that elements of $\mathbb{P}_+(S)$ are unordered sets; we define an ordered analogue as follows:

$$\mathbb{O}_+(S) := \coprod_{n \geq 1} S^n,$$

where $S^n = S \times S \times \cdots \times S$ is the set of $n$-tuples in $S$. Each tuple $x \in \mathbb{O}_+(S)$ has a cardinality $k = |x|$, where $x \in S^k$; we may refer to $x$ as a $k$-edge. There is an obvious map $\mathbb{O}_+(S) \to \mathbb{P}_+(S)$ given by sending each $n$-tuple to its set of elements; note that this map may decrease cardinality.

**Definition 2.3.** A directed hypergraph is a triple $X = (X_E, X_V, f)$, where $X_E$ and $X_V$ are sets (called the set of hyperedges and vertices of $X$, respectively), and where $f : X_E \to \mathbb{O}_+(X_V)$ is a function, which we call the constituent function.

An undirected hypergraph is a triple $X = (X_E, X_V, f)$ with $X_E$ and $X_V$ as above, and where $f : E \to \mathbb{P}_+(V)$ is a function, again called the (unordered) constituent function.

We can see easily that a directed graph is just a directed hypergraph with no $k$-edges for $k \neq 2$. This generalizes as follows.

**Definition 2.4.** A $k$-uniform hypergraph (or simply $k$-graph) is a hypergraph $(X_E, X_V, f)$, such that for all $e \in E$ the cardinality $|e| = k$.

The above definition is meant to include both directed and undirected hypergraphs, using the appropriate definitions of cardinality given above; again a (directed or undirected) 2-uniform graph is just a graph.

There is a well-defined notion of morphisms (or functions) between graphs (resp. hypergraphs), which we give below. These morphisms tell us what structure needs to be preserved when comparing two graphs (resp. hypergraphs). The idea is that one should send vertices to vertices and edges to edges, in a consistent way.

In the following definition and several others, we will assume the reader knows the first few definitions of category theory, which can be found in [?]. One thinks of a category as roughly like a directed graph, the main difference being a certain transitivity law: every pair of edges $a \to b$ and $b \to c$ gives rise to a distinguished composition edge $a \to c$. The nodes in a category are called objects and the edges are called morphisms. Note that there is a layer of abstraction here: if a category is like a graph, then the category of hypergraphs is like a graph that has hypergraphs as nodes and functions between hypergraphs as edges.

**Definition 2.5.** Let $X = (X_E \xrightarrow{f} X_V \times X_V)$ and $Y = (Y_E \xrightarrow{g} Y_V \times Y_V)$ denote graphs. A morphism $X \to Y$ consists of a function on edges $a : X_E \to Y_E$ and a function on vertices $b : X_V \to Y_V$, such that the diagram

$$
\begin{array}{ccc}
X_E & \xrightarrow{f} & X_V \times X_V \\
\downarrow{a} & & \downarrow{b \times b} \\
Y_E & \xrightarrow{g} & Y_V \times Y_V
\end{array}
$$

commutes.

**Definition 2.6.** Let $X = (X_E, X_V, f)$ and $Y = (Y_E, Y_V, g)$ be directed hypergraphs. A morphism $X \to Y$ consists of functions $a : X_E \to Y_E$ and $b : X_V \to Y_V$,
such that the diagram
\[
\begin{array}{c}
X_E \xrightarrow{f} \mathbb{O}(X_V) \\
\downarrow a \quad \downarrow b \\
Y_E \xrightarrow{g} \mathbb{O}(Y_V)
\end{array}
\]
commutes. Here \(b_*(v_1, \ldots, v_n) = (b(v_1), \ldots, b(v_n))\).

If \(X\) and \(Y\) are undirected hypergraphs, we define morphisms \(X \to Y\) in the same way, except that in the diagram we replace \(\mathbb{O}_+\) with \(\mathbb{P}_+\) and say that \(b_*(\{v_1, \ldots, v_n\}) = \{b(v_1), \ldots, b(v_n)\}\).

The category of directed (resp. undirected) hypergraphs has now been defined; we denote it \(\mathcal{H}\).

2.7. Semi-simplicial sets. We discussed simplicial sets in the introduction, but everything we said there could be said about semi-simplicial sets as well. After giving the definition (2.8) of semi-simplicial sets, we discuss their geometric visualization (which can be made precise) in Remark 2.9. We will briefly explicate the differences between simplicial and semi-simplicial sets in Remark 2.19.

Definition 2.8 (Set-theoretic definition of semi-simplicial set). A semi-simplicial set \(X\) consists of a sequence of sets \((X_0, X_1, X_2, \ldots)\), and for every \(n \geq 1\) a collection of functions \(d^n_0, d^n_1, \ldots, d^n_n\) each of which has domain \(X_n\) and codomain \(X_{n-1}\). These functions must satisfy the following identity:
\[
d^n_i \circ d^n_{j+1} = d^n_{j-1} \circ d^n_{i+1}, \quad \text{for all } 0 \leq i < j \leq n.
\]

For each \(n \geq 0\) the set \(X_n\) is called the set of \(n\)-simplices of \(X\). The function \(d^n_i : X_n \to X_{n-1}\) is often abbreviated \(d_i\) and is called the \(i\)th face operator. The dimension of \(X\) is the largest number \(n\) such that \(X_n \neq \emptyset\), if a largest such \(n\) exists, and \(\infty\) otherwise.

We may write \(X\) as
\[
\cdots \to X_2 \xrightarrow{d_1} X_1 \xrightarrow{d_0} X_0.
\]

Remark 2.9. Definition 2.8 may seem difficult to comprehend on first glance. One should think of \(X\) as a polygonal shape made of triangles, like two triangles put together to form a square (see Example 2.11). The set \(X_0\) is the set of vertices of that shape, the set \(X_1\) is the set of edges, the set \(X_2\) is the set of triangles (called 2-simplices in this parlance because they are 2-dimensional), etc.

Each 2-simplex in \(X\) has three edges: across from each vertex in the triangle is an edge not touching it. The three face operators \(d^2_0, d^2_1, d^2_2\) are tasked with telling us which three edges form the boundary of a given 2-simplex.

The above visualization can be made precise, in the form of geometric realization, which is a functor from the category of semi-simplicial sets to the category of topological spaces. See [?].

Definition 2.10. Let \(X = \cdots \xrightarrow{d_2} \xrightarrow{d_1} \xrightarrow{d_0} X_0\) be a semi-simplicial set. The 1-skeleton of \(X\) is the (directed) graph
\[
X_1 \xrightarrow{d_1} \xrightarrow{d_0} X_0,
\]
or one could write \(X_1 \xrightarrow{d_1 \times d_0} X_0 \times X_0\).
obtained by forgetting all higher-order \((n \geq 2)\) data in \(X\).

**Example 2.11.** A filled-in square, composed of two solid triangles glued along their diagonal, is drawn below.

Here we have \(X_0 = \{a, b, c, d\}, X_1 = \{ab, ac, ad, bd, cd\}, X_2 = \{abd, acd\}\), and \(X_n = \emptyset\) for \(n \geq 3\). Each triangle has three edges, given by the various \(d^2\)'s: \(d^2_0(abd) = bd, d^2_0(acd) = cd, d^2_1(abd) = d^1_1(acd) = ad, d^2_2(abd) = ab,\) and \(d^2_3(acd) = ac\). Similarly, each edge has two vertices: \(d^1_1(ac) = d^1_1(ad) = d^1_1(ab) = a, d^0_1(ad) = d^0_1(bd) = d^0_1(cd) = d, d^0_1(ac) = d^0_1(cd) = c,\) and \(d^0_0(ab) = d^0_1(bd) = b\).

Knowing the above sets \(X\) and face operator equations allows one to completely reconstruct the square. They are precisely the data of the semi-simplicial set \(X\).

The 1-skeleton of the above square is drawn the same way, but one imagines the triangles as empty in this case; it is just four vertices and five edges. As a graph, it is written \(\{ab, ac, ad, bd, cd\} \rightarrow \{a, b, c, d\}\), where the map assigns to each edge the pair consisting of its first and second vertex. This is precisely what is achieved by \(d^1_1\) and \(d^0_1\) above.

The set-theoretic definition of simplicial sets is much more involved than that of semi-simplicial sets, the difference being that \(n\)-simplices in a simplicial set can be “degenerate,” meaning they are really smaller simplices in disguise. Instead of having to satisfy one identity (like the displayed formula (1)), a simplicial set must satisfy six. For this reason, we do not include the set-theoretic definition of simplicial sets; it can be found in [\textcopyright{}]. Luckily, the categorical definition of simplicial sets is no harder than that of semi-simplicial sets. Compare Lemma 2.15 and Definition 2.17.

2.12. Categorical definitions of graphs, hypergraphs, and simplicial sets.

In this section we give category-theoretic definitions of graphs, hypergraphs, semi-simplicial sets, and simplicial sets. We find that there is a common category-theoretic framework which holds all three examples (and more, as we will see in the next section). That is, we can consider each of them as a *category of functors*, from some “indexing” category to the category of sets. In the case of graphs the indexing category will have two objects, \(V\) and \(E_2\), and two maps \(V \rightarrow E_2\); in the case of hypergraphs the indexing category will have (infinitely-many) objects \(V, E_1, E_2, \ldots\) and \(n\) maps \(V \rightarrow E_n\) for each \(n\); and in the case of semi-simplicial sets the indexing category will have objects \([0], [1], [2], \ldots\) and \(\binom{j+1}{i+1}\) maps \([i] \rightarrow [j]\).

Recall that if \(\mathcal{C}\) is a category then a functor \(F: \mathcal{C} \rightarrow \text{Sets}\) assigns to each object \(C \in \text{Ob}(\mathcal{C})\) a set \(F(C)\), and to each morphism \(C \rightarrow D\) in \(\mathcal{C}\) a function between sets \(F(C) \rightarrow F(D)\). If \(G: \mathcal{C} \rightarrow \text{Sets}\) is another functor, then a natural transformation \(\alpha: F \rightarrow G\) is a coherent way to give a function \(F(C) \rightarrow G(C)\) for each \(C \in \text{Ob}(\mathcal{C})\). See [\textcopyright{}].

The functors \(\mathcal{C} \rightarrow \text{Sets}\) are thus themselves the objects of a larger category, written \(\text{Sets}^{\mathcal{C}}\), where the morphisms in this category are the natural transformations between functors. While this may be difficult to understand at first, one should remember that we are merely indexing sets and functions. The category-theoretic
definition of graphs, given as Lemma 2.13 below, may help orient the reader to this way of thinking.

**Lemma 2.13 (Categorical definition of graphs).** Let $\Delta_G$ denote the category with two objects $V$ and $E_2$, two morphisms $V \to E_2$, and no other non-identity morphisms. Let $\Delta_G^{op}$ denote its opposite. Then the category of graphs is equivalent to the category of functors $\textbf{Sets}^{\Delta_G^{op}}$.

Proof. Giving a functor $F: \Delta_G^{op} \to \textbf{Sets}$ consists of giving two sets, $F(V)$ and $F(E_2)$, and two functions $F(E_2) \to F(V)$. This is precisely the data of a graph $F_E \to F_V \times F_V$ with vertex set $F_V := F(V)$ and edge set $F_E := F(E_2)$. Given a natural transformation $F \to F'$ consists of giving a function $F(V) \to F'(V)$ and a function $F(E_2) \to F'(E_2)$, such that the diagrams

$$
\begin{array}{ccc}
F(E_2) & \longrightarrow & F(V) \times F(V) \\
\downarrow & & \downarrow \\
F'(E_2) & \longrightarrow & F'(V) \times F'(V)
\end{array}
$$

commute. Such is precisely a morphism of graphs.

□

**Lemma 2.14 (Categorical definition of hypergraphs).** Let $\Delta_H$ denote the category with objects denoted $V, E_1, E_2, E_3, \ldots$ and with distinct morphisms $v_i^n: V \to E_n$ for each $1 \leq i \leq n$, and no other non-identity morphisms. Let $\Delta_H^{op}$ denote its opposite. The category $\mathcal{H}$ of directed hypergraphs is equivalent to $\textbf{Sets}^{\Delta_H^{op}}$, the category of contravariant functors from $\Delta_H$ to $\textbf{Sets}$.

Sketch of proof. Let $F: \Delta_H^{op} \to \textbf{Sets}$ denote an object in $\textbf{Sets}^{\Delta_H^{op}}$. As a directed hypergraph, its vertex set is $F_V := F(V)$, its hyperedge set is $F_E := \bigsqcup_{n \in \mathbb{N}} F(E_n)$, and its constituent function is given by the product of the $n$ maps $F(v_i^n): F(E_n) \to F(V)$. One checks that a natural transformation $F \to F'$ is the same data as the respective morphism of hypergraphs.

□

As promised, we also give a category-theoretic definition of semi-simplicial sets. For each $n \in \mathbb{N}$, let $[n]$ denote the ordered set $\{0, 1, \ldots, n\}$; note that each $[n]$ is finite and non-empty. A function $f: [m] \to [n]$ is called non-decreasing if $j \geq i$ implies $f(j) \geq f(i)$, for all $0 \leq i, j \leq m$. The function $f$ is called strictly increasing if $j > i$ implies $f(j) > f(i)$. The category whose objects are the ordered sets $[n]$ for $n \in \mathbb{N}$ and whose morphisms are non-decreasing (respectively strictly-increasing) functions is denoted $\Delta_\geq$ (respectively $\Delta_>$).

**Lemma 2.15 (Categorical definition of semi-simplicial sets).** Let $\Delta_>$ denote the category whose objects are the finite non-empty ordered sets $\{0, 1, \ldots, n\}, n \geq 0$, and whose morphisms are strictly increasing maps. Let $\Delta_\geq^{op}$ denote the opposite category. The category of semi-simplicial sets is the category of functors $\textbf{Sets}^{\Delta_\geq^{op}}$, whose objects are functors $\Delta_\geq^{op} \to \textbf{Sets}$ and whose morphisms are natural transformations.

**Remark 2.16.** Given a semi-simplicial set $X: \Delta_\geq^{op} \to \textbf{Sets}$, we write $X_n \in \textbf{Sets}$ to denote the set $X(\{0, 1, \ldots, n\})$; its elements are the $n$-simplices of $X$. There are
n + 1 possibilities for strictly increasing maps $d^i: \{0, 1, \ldots, n - 1\} \to \{0, 1, \ldots, n\}$, given by skipping exactly one of the numbers $0 \leq i \leq n$. Since $X$ is a contravariant functor, each of these morphisms is sent to a map $X(d^i): X_n \to X_{n-1}$, which is the $i^{th}$ face operator and denoted $d^i_n$. These face operators satisfy the properties given in the set-theoretic Definition 2.8.

**Definition 2.17** (Categorical definition of simplicial sets). Let $\Delta_{\geq}$ denote the category whose objects are the finite non-empty ordered sets $\{0, 1, \ldots, n\}, n \geq 0$, and whose morphisms are non-decreasing maps. Let $\Delta_{\geq}^{op}$ denote the opposite category. The category of simplicial sets, denoted $sSets$, is the category of functors $\Delta_{\geq}^{op} \to Sets$, whose objects are functors $\Delta_{\geq}^{op} \to Sets$ and whose morphisms are natural transformations.

**Remark 2.18.** This is not a technical paper designed to prove difficult results, but instead an attempt to convince the reader that simplicial sets are better models for certain networks. Hence, we do not require that the reader understand the rigorous underpinnings of simplicial sets. If convinced of their usefulness, the reader can easily learn about them on his or her own time. See [?] for a nice exposition. We hope the reader simply understands the following.

A simplicial set is a geometric object with corners, edges, triangular faces, tetrahedra, etc., which are called simplices. An $n$-simplex has dimension $n$, so that a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and an $n$-simplex is an $n$-dimensional triangle. Each $n$-simplex has lots of sub-simplices, as a triangle has edges and vertices. All simplices have ordered vertices, and the vertices of a sub-simplex are in the same order as in the simplex.

If simplicial sets are geometric objects comprised of these triangles, then we must be able to fit together these triangles in some ways. To do so, simply line up some vertices of one simplex with some vertices of another; doing so will match up a face of the first simplex with a face of the other. Then attach these two simplices along that common face. In Example 2.11 we showed a square; it can be obtained by an attaching process as we have described here, by gluing the triangle $abd$ to the triangle $acd$ along the common edge $ad$.

**Remark 2.19.** We promised to compare simplicial sets and semi-simplicial sets. These notions have had a long history, and it took many years to sort out terminology (see [?]). Both are useful, although simplicial sets have become dominant in algebraic topology.

To compare semi-simplicial sets and simplicial sets, one should begin by comparing their “indexing categories,” $\Delta_{\geq}$ and $\Delta_{\geq}$, the difference being the inclusion of non-injective morphisms in $\Delta_{\geq}$. Geometrically, this allows one to consider simplices (triangles) in a simplicial set that are degenerate. A degenerate $n$-simplex is an $n$-dimensional triangle that has been flattened onto one of its $(n-k)$-dimensional faces. The use of these degenerate simplices has two advantages from a topological perspective: it is easier to build most shapes using simplicial sets (e.g. the 2-sphere), and products of simplicial sets are better behaved under geometric realization (see [?]).

2.20. **Comparing hypergraphs and simplicial sets.** We have found that graphs, hypergraphs, and (semi-) simplicial sets are all examples of the same phenomenon: each is a category of functors from some indexing category to $Sets$, meaning each is simply a prescribed arrangement of sets. This discovery leads to an increased
ability to change between one model and another; namely we just need to find a relationship between the categories that index them. Let us make this more precise.

**Lemma 2.21.** Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Given a functor $A: \mathcal{C} \to \mathcal{D}$, one obtains a functor $R_A: \mathbf{Sets}^D \to \mathbf{Sets}^C$ given by $R_A(G) = G \circ A$, for any $G \in \mathbf{Sets}^D$. There is also a functor $L_A: \mathbf{Sets}^C \to \mathbf{Sets}^D$, and a natural bijection

$$\text{Hom}_{\mathbf{Sets}^C}(F, R_A(G)) \cong \text{Hom}_{\mathbf{Sets}^D}(L_A(F), G)$$

for any $F \in \mathbf{Sets}^C, G \in \mathbf{Sets}^D$.

**Proof.** See [?], Thm I.5.2.

To bring this down to earth, recall the categories $\Delta_G$ and $\Delta_H$ from Lemmas 2.13 and 2.14. There is an obvious functor $\Delta_G \to \Delta_H$ sending $V \mapsto \{0\}$ and for each $n \geq 1$, send $E_n \mapsto [n-1]$. On morphisms, send $v^i_n: V \to E_n$ to the morphism $(0 \mapsto i-1): \{0\} \to [n-1]$, for each $1 \leq i \leq n$. Let $A: \Delta_H^\text{op} \to \Delta_G^\text{op}$ denote its opposite. By Lemma 2.21, we obtain functors

$$\mathcal{G} \xrightarrow{L_i} \mathcal{H} \xleftarrow{R_i} \mathcal{G}.$$

The functor $R_i$ takes a hypergraph and forgets everything but the vertices and the 2-edges, to yield a graph. The functor $L_i$ takes a graph and considers it as a (2-uniform) hypergraph. Note that $R_i \circ L_i: \mathcal{G} \to \mathcal{G}$ takes any graph to itself, but the functor $L_i \circ R_i: \mathcal{H} \to \mathcal{H}$ destroys information.

We move on to compare hypergraphs and simplicial sets in the same way, by supplying a functor between indexing categories $\Delta_H \to \Delta_\geq$ as follows. On objects, send $V \mapsto \{0\}$, and for each $n \geq 1$, send $E_n \mapsto [n-1]$. On morphisms, send $v^i_n: V \to E_n$ to the morphism $(0 \mapsto i-1): \{0\} \to [n-1]$, for each $1 \leq i \leq n$. Let $A: \Delta_H^\text{op} \to \Delta_\leq^\text{op}$ denote its opposite. By Lemma 2.21, we obtain functors

$$\mathcal{H} \xrightarrow{L_A} \mathbf{csSets} \xleftarrow{R_A} \mathcal{H}.$$

The functor $R_A$ takes a simplicial set and forgets all relationships between $n$-simplices and $m$-simplices, except if $m = 0$. The functor $L_A$ takes a hypergraph $H$ and sends it to a simplicial set whose 0-simplices are the vertices of $H$, whose $n$-simplices are the $(n+1)$-edges of $H$ for $n \geq 1$. Note that the composition $R_A \circ L_A: \mathcal{H} \to \mathcal{H}$ is not quite the identity: a hypergraph is replaced by the one in which every vertex is considered a 1-edge (and everything else stays the same). Some authors ([?]) define hypergraphs as not having 1-edges to begin with; for them the composite $R_A \circ L_A$ would be the identity.

Similarly, there are functors between simplicial sets and semi-simplicial sets, between graphs and (semi-) simplicial sets, etc. Any functor between indexing categories gives rise to functors between the categories they index. This is the basic idea behind the next subsection, where we define what it means to be a category of higher graphs.

### 2.22. Categories of higher graphs.

We saw above that category theory provides a framework in which graphs, hypergraphs, simplicial sets, and semi-simplicial sets all can be defined. One naturally asks: what other categories might be useful for modeling networks?
In order to answer this, we must decide on what we want out of a theory of higher-dimensional graphs should have. We clearly want a set of vertices and sets of (higher) edges, such that each higher edge has an underlying set of vertices. It seems that having a set of 1-edges is not essential, so we do not require it.

**Definition 2.23.** Let $\Delta_B$ denote the category with an object $V$ and objects $\{E_n|n \geq 2\}$, and with morphisms $e^n_i: V \to E_n$ for each $1 \leq i \leq n$. We call $\Delta_B$ the basic indexing category for higher graphs (or just basic indexing category for short).

An indexing category for higher graphs (or indexing category for short) is a pair $(\mathcal{I}, \Delta_B \xrightarrow{\mathcal{I}} \mathcal{I})$, where $\mathcal{I}$ is a category and $f: \Delta_B \to \mathcal{I}$ is a functor. We may denote $(\mathcal{I}, \Delta_B \xrightarrow{\mathcal{I}} \mathcal{I})$ simply by $\mathcal{I}$, if $f$ is clear from context.

We say that an indexing category $(\mathcal{I}, \Delta_B \xrightarrow{\mathcal{I}} \mathcal{I})$ is faithful if $f$ is a faithful functor.

**Remark 2.24.** An indexing category for higher graphs is just a category which has an “interpretation” of vertices and $n$-edges for each $n \geq 2$. It is faithful if the $n$ vertices of an $n$-edge are “distinct.” Thus each of $\Delta_H, \Delta_S$, and $\Delta_>$ is a faithful indexing category for higher graphs in an obvious way. In fact, even $\Delta_G$ is an indexing category for higher graphs (in many different ways), because there are infinitely many functors $\Delta_B \to \Delta_G$. However, none of these are faithful, so $\Delta_G$ is cannot be made a faithful indexing category for higher graphs.

The above paragraph focuses on indexing categories for higher graphs. After making the following definition (2.25), we will be able to say that hypergraphs, semi-simplicial sets, and simplicial sets are all faithful categories of higher graphs.

**Definition 2.25.** The category of functors $\Delta^{op}_B \to \text{Sets}$, denoted $\mathcal{B} = \text{Sets}^{\Delta^{op}_B}$, is called the basic category of higher graphs. The objects of $\mathcal{B}$ are called basic higher graphs (or just basic graphs for short).

A category of higher graphs (or just category of graphs for short) is a category of the form $\text{Sets}^{\mathcal{I}}$, where $(\mathcal{I}, \mathcal{B} \xleftarrow{\mathcal{I}} \mathcal{I})$ is an indexing category for higher graphs. We say that it is faithful if $f$ is. We call $\text{Sets}^{\mathcal{I}}$ the category of $\mathcal{I}$-indexed graphs.

Given an indexing category $(\mathcal{I}, \mathcal{B} \xleftarrow{\mathcal{I}} \mathcal{I})$ there is always a pair of functors

$$
\begin{array}{ccc}
\mathcal{B} & \xrightarrow{L_f} & \text{Sets}^{\mathcal{I}} \\
& \xleftarrow{R_f} & \\
\end{array}
$$

in accord with Lemma 2.21, comparing the category of $\mathcal{I}$-indexed graphs and the category of basic graphs. For any $\mathcal{I}$-indexed graph $G: \mathcal{I}^{op} \to \text{Sets}$, we call $R_f(G)$ its underlying basic graph.

**Example 2.26 (Undirected higher graphs).** Let $\Sigma$ denote the category whose objects are the sets $\underline{n} := \{0, 1, \ldots, n\}$, one for each $n \in \mathbb{N}$, and whose morphisms are simply functions. Thus there are $(n+1)(n+1)$ morphisms $m \to \underline{n}$. Note that $\Sigma$ is a faithful indexing category for higher graphs via the obvious functor $\mathcal{B} \to \Sigma$ sending $V \to \underline{0}$ and $E_n \to \underline{n-1}$ for $n \geq 2$. The category $\text{Sets}^{\Sigma^{op}}$ of $\Sigma$-indexed graphs may also be called the category of symmetric simplicial sets. See [?].

One can visualize a symmetric simplicial set as a simplicial set in which the vertices are unordered. It is the union of vertices, edges, triangles, etc. Symmetric simplicial sets are the higher-dimensional analogue of undirected graphs.
Hypergraphs underlying symmetric simplicial sets have the property that if $e$ is an $n$-edge then there are at least $n!$ edges with the same vertices as $e$ in the permuted orders.

**Example 2.27 (Broadcasting).** For any $n \in \mathbb{N}$, let $(n)$ denote the category with an initial object $Q_n$ and $n$ other objects, and that has no non-identity morphisms except for the unique morphism from $Q_n$ to each other object. Note that there are $(n+1)^n + n$ functors $(m) \to (n)$. Let $T$ denote the category whose objects are the categories $(n)$ and whose morphisms are the functors between them. We call $T$ the *indexing category for broadcasts*.

There is an obvious functor $B \to T$ given by $V \mapsto (0)$ and $E_n \mapsto (n-1)$ for each $n \geq 2$. In fact this functor is faithful, so $T$ is a faithful indexing category for higher graphs.

One may think of a $T$-indexed graph $F : T^{\text{op}} \to \text{Sets}$ as a model for broadcasting and rebroadcasting. The set $T(n)$ is the set of broadcasts that reach $n$ other entities. Morphisms $(m) \to (n)$ allow one to connect various broadcasts together to form a network. Experts may also wish to use a rather obvious Grothendieck topology on $T$ and then work with sheaves rather than presheaves.

For completeness, we also define a morphism of indexing categories.

**Definition 2.28.** Let $(\mathcal{I}, B \xrightarrow{f} \mathcal{I})$ and $(\mathcal{J}, B \xrightarrow{g} \mathcal{J})$ denote indexing categories for higher graphs. A *morphism of indexing categories* is functor $h : \mathcal{I} \to \mathcal{J}$ such that $g \circ h = f$.

### 3. Choosing the appropriate model

As discussed in the introduction, graphs are not suitable models for many kinds of networks that exist in the real world. One way of stating the inadequacy is that graphs do not contain enough data to discern multi-person conversations from private ones. They do not leave room for contexts shared by more than two people.

For example, 3-way calling or conference calls cannot be adequately modeled by graphs. Information is transferred differently in a 3-way call between $A, B,$ and $C$ than it is if the three people can only talk in two-way conversations. The world of conference calls can be modeled by hypergraphs, with an edge connecting any set of vertices that are engaged in a given call. It would not be wrong to model this with situation with simplicial sets, but doing so will not really contribute anything of value. However, one may imagine adding a feature to conference calls in which any person can, if he or she chooses, speak more privately to any subset of the others. This second scenario is nicely modeled by simplicial sets (in the case of our 3-way call, it is modeled by a 2-simplex with vertices $A, B, C$).

Imagine that an intelligence agency (like the CIA) is trying to listen to the above three-way conversation, but for some reason they can only listen to the communication between $B$ and $C$. This scenario is modeled by a simplicial set with two 2-simplices, $ABC$ and $BCI$, where $I$ represents the intelligence agency. These two 2-simplices are glued together along the common 1-simplex $BC$. Such a scenario cannot be adequately modeled by hypergraphs, because of the “sub-edge problem” detailed below in Remark 3.5.

In this section, we try to make clear how one should choose whether to model a given scenario with graphs, hypergraphs, simplicial sets, symmetric simplicial sets, etc. Clearly, one wants to choose the model whose indexing category is as simple as
possible, under the condition that the model accurately reflects all of the features that one wishes to study. Note however that if simplicial sets can be used, they may provide extra features that, say, hypergraphs do not. Namely, the fact that simplicial sets can be visualized as topological spaces and analyzed as such with sophisticated algebraic tools may render the simplicial model more useful.

3.1. Networks modeled by graphs. Many networks can be modeled by (1-dimensional) graphs. When a person sends email, it always passes privately to the recipient. Even if the recipient list of an email includes multiple addresses, none of the recipients has any way to truly know that the same message was sent to each person. The message does not exist in a shared space.

Graphs are good models for any type of network in which all correspondences are private, or “one-on-one,” because every incidence between nodes involves only two nodes. The graph model breaks down as soon as one wants to consider threesome, foursomes or n-somes interacting all at once. In an online social network (e.g. Facebook), some communications are posted to a public forum which can be accessed by any member of the group. Assuming the internal integrity of the system (i.e. assuming Facebook displays the same messages to each member of the group), the message is shared by the group. These situations are not well-modeled by graphs.

The following are some networks that are best modeled with graphs (§).

1. transportation networks (locations, roads);
2. tournaments (players, one-on-one matches);
3. electrical flow (terminals, wires);
4. private gossip (people, one-on-one conversations).

What these networks have in common is that every interaction is between exactly two nodes – the words in parentheses correspond to the vertices and edges in the graphical model.

3.2. Networks modeled by hypergraphs. In many applications, there may be groups that interact in a way that subgroups do not; hypergraphs are often appropriate models for these situations. As we mentioned above, in a conference call any person can speak to the whole group but cannot speak to a given subgroup. In Facebook, one may post a message to an entire group of which he or she is a part, but not to a specified subgroup.

As another example, one may model the academic research network by a hypergraph in which every researcher is a vertex, and every coauthorship is a hyper-edge. If three researchers are the coauthors on a paper, that does not imply that some subset of them is also the coauthors of a paper. Coauthorship is a relation that is not “closed under taking subsets” (see Definition 3.3). For this reason hypergraphs are the appropriate model of coauthorship. Similarly, the network consisting of sets of people who have eaten lunch as a group should be modeled by hypergraphs, because this relation is not closed under taking subsets.

Definition 3.3. A symmetric multi-relation on a set $S$ consists of a subset $R \subseteq \mathcal{P}(S)$, i.e. a set of subsets of $S$. We that $R$ is closed under taking subsets if, whenever $x \subseteq S$ is an element of $R$ and $x' \subseteq x$ is a subset of it, then $x' \in R$ as well.

Remark 3.4. There is also an analogous notion of (non-symmetric) multi-relation $R \subseteq \bigcup_{n \in \mathbb{N}} S^n$, and of such a multi-relation being closed under taking subsets, but we will not define the latter here.
We can see that hypergraphs are good models for multi-relations. If the multi-relation is closed under taking subsets, then simplicial sets (or symmetric simplicial sets, etc.) can also be used. It is often useful to change ones perspective on a relationship so that it becomes closed under taking subsets. Let us show what we mean by taking another look at the examples above.

Instead of modeling a “coauthorship” network, change perspectives and model a “collaboration” network: if \( n \) people have all worked together on a project then so has every subgroup. Or, instead of looking at the network of sets of people who have had lunch as a group, look at the network of people who have had lunch together. The same goes for conference calls and Facebook groups: if a group of people are involved together, then so is any subgroup.

In all of these cases, one can change perspectives on what the network is, and then use simplicial sets as a model. The maximal simplices in the simplicial set will correspond to the hyperedges of the original hypergraph (that is, we can always recover the hypergraph from the simplicial set). The simplicial model may even suggest another feature. For example, as mentioned above, one might want to add a feature to conference calling (or to Facebook groups) whereby a participant can speak privately to a subset of the other participants.

Connecting groups together is where simplicial sets have a big advantage over hypergraphs. If two members of a collaborative effort take their knowledge and apply it to another collaborative effort, we may want to imagine this as two simplices joined together along a common edge in the collaboration network. We see show below that hypergraphs have a serious drawback in this context, which we have not yet made explicit. We call it the sub-edge problem.

**Remark 3.5 (Sub-edge problem).** Suppose that \( X \) is a hypergraph with vertices \( \{a, b, c, d\} \) and three edges \( e_1 = (a, b), e_2 = (a, b, c), e_3 = (a, b, c) \), where two edges have the same vertices. If one were to ask whether \( e_1 \) is a subedge of \( e_2 \) or \( e_3 \), either we would decline to answer (in Japanese we would say \( mu \): “unask the question”), or we would be required to say that \( e_1 \) is a subedge of both \( e_2 \) and \( e_3 \). Using hypergraphs, there is no good notion of sub-edges.

This issue is clear when looking at the underlying indexing category for hypergraphs, \( \Delta_H \). It has no morphisms \( E_n \to E_m \) for any \( m \neq n \). Thus a hypergraph \( X: \Delta_H^{op} \to \text{Sets} \) has no capability for comparing the sets \( X(E_m) \) of \( m \)-edges and \( X(E_n) \) of \( n \)-edges.

We explain the problem from one more angle. Suppose that \( X' \) is the hypergraph with the same vertices as \( X \) but with a slight change in the edge sets: \( e'_1 = (a, b), e'_2 = (b, c), e'_3 = (a, b, c) \). There is no morphism \( X' \to X \) respecting the vertices, even though \( e'_2 \) looks like a subedge of \( e_2 \).

As an application, suppose we are modeling groups in Facebook, for example, and user \( a \) joins group \( e'_2 \). This event cannot be modeled by a morphism of hypergraphs.

Hypergraphs have no good notion of sub-edges, but simplicial sets do (have a good notion of sub-simplices). Thus all of the above issues go away when dealing with simplicial sets, or any model of higher graphs whose indexing category has an adequate set of morphisms.

3.6. **Network modeled by simplicial sets.** If we look back at the definition of simplicial sets (Definition 2.17), we notice that each simplex has ordered vertices. In many of our discussions above, we have been suppressing this fact. Simplicial sets may “look like” the polygonal shapes we’ve described, but they also have more
structure. This is akin to the fact that graphs can be directed; they may look like 1-dimensional spaces, but each edge has ordered vertices, giving it a directionality.

To represent a given real-world situation, this ordered structure may be useful or it may not. If it is not, we could either continue to use simplicial sets as our model but forget the ordered structure, or we could use symmetric simplicial sets (see Example 2.26). In this subsection we discuss situations in which simplicial sets, in all their ordered glory, are more appropriate as models than their symmetric analogues are. In Subsection 3.8 we explain situations in which other models are more appropriate.

We begin again with 1-dimensional simplicial sets, because these are just graphs; in this case, we are just asking how one decides when directed graphs are better models than undirected graphs are. One should think of undirected graphs as symmetric directed graphs, i.e. directed graphs $G$ such that if $v,v'$ are vertices, then there is a bijection between the set of directed edges connecting $v$ to $v'$ and the set of directed edges connecting $v'$ to $v$. With this condition imposed, there is no need to draw the arrows.

Undirected graphs are not capable of expressing as much as directed graphs are. However, this extra expressiveness for directed graphs has its drawbacks. First, they take up more space and are harder to analyze because they have more structure. Second, that extra structure is sometimes not relevant to the task at hand: if all interactions in a certain arena are by nature symmetric, then adding the extra structure of directionality is not only slower, it’s misrepresentative of the inherent structure of the given arena. Category theory teaches us to take these types of considerations seriously. However, if interactions in the network naturally have an order (like transportation from A to B), then we use directed graphs as models.

Similarly, the ordering on simplices would be useful for modeling a network if, within any given interaction of that network, the nodes naturally have an order! For example, imagine a multi-cast network in which one node sends a message to a list of other nodes, with the provision that every node in the list is also responsible for the successful delivery of that message to all subsequent nodes. This scenario is best modeled by simplicial sets because the nodes in each simplex are ordered.

As another example, suppose that for every military mission, the soldiers involved are strictly ranked, so that each soldier either outranks or is outranked by each other soldier. A scenario in which three soldiers $A,B,C$ are simultaneously involved in two different missions, one consisting of five people and the other consisting of six people, would be modeled by the union of a 4-simplex and a 5-simplex, glued together along a common 2-simplex ($ABC$). The only condition is that the rank of $A, B,$ and $C$ is the same in both missions.

As a final example, consider a set of interacting processes, say in a computer program. Suppose that each process has a finite number $n$ of parts which should occur in a fixed order, but that some processes $P$ may be subprocesses of two distinct processes $Q, R$. We would model such a scenario as a simplicial set (in this case with $Q$ and $R$ glued together along $P$).

Just like directed graphs can be used to model undirected situations (one includes edges in both directions for every pair of connected nodes), simplicial sets can be used to model situations that are not inherently ordered. Again, one simply includes all $(n + 1)!$ permutations of any given $n$-simplex.
Simplicial sets are expressive and yet combinatorial (i.e. they fall under the banner of “discrete mathematics”). We record their wide-ranging mathematical application in the following remark.

**Remark 3.7.** Simplicial sets are widely used in category theory, algebra, and algebraic topology. In category theory, any monad on a category gives rise to a natural simplicial resolution of objects in that category. Thus simplicial sets come up as resolutions of objects in an algebraic category such as groups or algebras, and hence are indispensable in homology theory [?]. They also are sufficient as models for topological spaces up to homotopy [?]. Finally, new work by Joyal [?] and by Lurie [?] shows that simplicial sets also provide a good model for higher categories.

These applications and others may also prove useful or interesting to a computer science researcher.

### 3.8. Other networks

Many networks involve communications between more than two people, but with some special requirements for comparing such communications. Each of these networks should be modeled by a higher graph, but the choice of indexing category is a bit of an art.

We mentioned hierarchical networks in Subsection [3.6] using military operations as one example. However, because the objects of $\Delta_>$ are strictly ordered sets, we had to assume that the participants in any operation are also strictly ranked in order to use simplicial sets as a model. If instead some participants are ranked equally, then we may choose a different model, i.e. a category of higher graphs with an indexing category whose objects are not so strictly ranked.

The issue is really the anti-symmetric law: $x \leq y$ and $y \leq x$ implies $x = y$. This holds for any two elements $i, j$ of $[n] \in \Delta_\leq$. Let us relax that restriction.

The category of non-empty finite linear preorders, denoted $\Delta_{PrO}$ has sets $X$ as objects, where $X$ has elements $x_1, \ldots x_n$ (for some $n \geq 1$), and an ordering $\preceq$ such that $i \leq j$ implies $x_i \preceq x_j$. This ordering is not required to be anti-symmetric, but note that as we have defined it, $\preceq$ is automatically reflexive and transitive. A morphism of these linear pre-orders is just a non-decreasing function. Note that $\Delta_>$ is a subcategory of $\Delta_{PrO}$; hence $\Delta_{PrO}$ can be given the structure of an indexing category for higher graphs.

The category of $\Delta_{PrO}$-indexed graphs is the richest category of higher graphs we have discussed so far. Its objects look like simplicial sets, except that vertices in any simplex are not strictly ordered. If, for example, three soldiers $A, B, C$ are ranked equally when working in one operation, but unequally in another, the simplices representing the two operations may be glued together along the ordered 2-simplex. However, if the three soldiers are ranked $A \prec B \prec C$ in one operation and $A \prec C \prec B$ in another, then the two operations cannot be glued.

There are infinitely many categories that can serve as indexing categories for higher graphs (see Definition [2.23]). One chooses the appropriate indexing category by determining the key features of how edges are compared. For example, in hypergraphs, edges are only compared by their vertices, and hence $\Delta_H$ has no (non-identity) morphisms except involving the object $V$.

### 4. Applications and further research

We have tried to convince the reader, especially in Section [3] of the applicability of simplicial sets and other categories of higher graphs to the study of networks.
There we discussed conference calls, collaboration networks in academic research, military missions, interlocking computer processes, and more. We give just two more applications in this section.

The first application involves natural languages and the problem of communication. In the world, each person has their own way of speaking (in linguistics, this is called their *idiolect*). Many cultures also have their own dialect: a pair of siblings may have special words or phrases whose full meaning only they understand, an academic discipline may have its own jargon, and the signers of a document may be forced to define certain words (e.g. “landlord”) in certain ways. Together, all of these dialects form a network.

What can we say about this network? We first note that if a group of people speak a common dialect, then so does every subgroup. Simplicial sets begin to emerge as a good possibility for the model. Note that a simplicial set may have two different simplices with the same set of vertices: what might this represent? It represents the possibility that there are two different contexts involving the same set of people, and that these two contexts demand different dialects.

The second application we wish to mention is related to the first. In computer science there are many different protocols that govern how information is transferred. Each time a message is sent, it conforms to any number of different protocols. Each computer program that sends information to others (even one that simply exports documents in, say, “rich text format”) adheres to a certain set of rules. If a set of computers share the ability to encode and decode each other’s messages, we would consider them to be sharing a protocol in this sense.

We define a *context* to be a set of protocols, and a *subcontext* to be a superset of protocols – the idea being that in a subcontext one can demand more. Nodes in a network of computers can thus be arranged according to the contexts in which they are participating. We write a simplex for each context (where subsimplices represent subcontexts). For example if $A, B, C$ are communicating in accordance with a set of protocols, $B$ and $C$ might be in fact communicating in accordance with a stricter set of protocols because their communications are also part of some other context $B, C, D, E$. This small network would be modeled by the union of a 2-simplex ($ABC$) and a 3-simplex ($BCDE$) along a common edge ($BC$).

All of the above noted examples must be researched more thoroughly, in order to prove that there are advantages to using simplicial sets (or other models) instead of hypergraphs. As another project, one might attempt to provide a systematic way to classify networks that emerge in the study of the real world, and determine which category of higher graphs is best suited for each one.

The author is particularly interested in modeling information transfer in a network. Consider a party, modeled as a simplicial set in which the vertices represent people, and in which $n$-simplices represent sets of $n + 1$ people who can hear each other. This simplicial set $P$ is well-suited to understand how messages are transferred at this party, but not how those messages are interpreted. But suppose we also had access to something like a “category of vocabularies,” where a morphism from one object to another represents how words in one person’s vocabulary are interpreted in another’s. In this case, there should be a functor from the category of simplicies of $P$ to the category of vocabularies. Messages, together with their interpretation, become information. Thus if we could make all this precise, we would have a nice model for information transfer in the network.