Questions and speculation on learning and cohomology, Version 3

Joshua Tan

April 23, 2018

Abstract

I have tried to formulate a draft of some things I have been thinking at the beginning of my doctorate, in order to be able to get some feedback. Any comments or corrections are more than welcome. In particular, I would be very grateful for concrete (possibly partial) answers, for opinions on whether questions are interesting or not interesting, and for advice on things to read which I am unaware of. I apologize in advance for being too brief and not defining things in many places. ~~I expect to revise this document as I get feedback and learn more.~~ Version 3 will be the last major update to this document, as I focus attention on my doctoral thesis. I am still very interested in obtaining feedback on the ideas expressed here!

Contents

> "For an answer which cannot be expressed, the question too cannot be expressed. The riddle does not exist. If a question can be put at all, then it can also be answered." – Wittgenstein

1 Introduction

The reason for putting artificial intelligence (AI) and geometry together in the first place is due to an intuition I had very early on in 2011 as a student in robotics: what AI needed was a systematic way of putting together logic (in the form of algorithms, proofs, and engineering design) with real-world data, and that the statistical algorithms popular in machine learning comprised only one class of options. Geometry, I hoped, could be another—after all, somehow there was a "logic" embedded in the geometry of space-time called the physical laws. More than the hope for new algorithms, however, I wanted to construct the same sort of mathematical semantics for AI as there exists for the theory of computation. I believed that the field of geometry could give *an organizing principle* for AI. By organizing principle I mean not just a classification of objects but some means of comparing and combining *tools and methods* in the discipline. Concretely, I wanted not just a classification of different mathematical models in AI (from ontologies to dynamical systems to Bayesian networks to simple, Boolean functions) but some means of comparing and combining "learning algorithms" meant to construct those models from data. The intuition for applying geometry is supported in part by the success of category theory in combining and composing tools and methods from many different areas of math, and in part by formal similarities between algebraic geometry and model theory (and now between homotopy theory and type theory) which illustrate how problems posed in logic and computer science could be transposed to cleaner frameworks of geometry, number theory, and category theory. Existing applications also show that such connections can be fruitful, from information geometry to persistent homology to geometric complexity to the use of topoi in categorial logic. In any case, AI is a jigsaw puzzle, and I need a way of organizing the pieces. Geometry seems like a good bet.

For this to work, first off I need to find some non-trivial "isomorphic" structures in AI and in geometry. Given such structures, it should be possible to apply organizing principles in geometry to AI. This hope is the subject of this brief essay. So far there isn't much understanding, but rather a list of things I would like to understand in the future.

1.1 How to read this essay

Sections 1-4 form the core narrative, reviewing developments across AI, algebraic topology, and algebraic geometry. Section 5 contains a few concrete proposals for future work. The lettered sections in the appendix discuss some particular research topics related to geometric intelligence and should be regarded as pointers to further reading.

1.2 Acknowledgements

I've had many interesting discussions in the course of this research. In particular, I would like to thank Samson Abramsky, Bob Coecke, David Spivak, Misha Gromov, Yiannis Vassolopoulos, Mehryar Mohri, Sylvain Cappell, and Brian Scassellati for help in developing some of these ideas.

This essay is based on a similar document [52] by Andreas Holmstrom on cohomology theories in arithmetic geometry.

1.3 Examples of connections between AI and geometry

To motivate the problem, we list some examples (in no particular order): invariant methods in computer vision, motion tracking and planning, configuration spaces in kinematics, foldable robots and surfaces, grid cell geometry in the hippocampus, differential geometry of neuronal networks, mobile sensor networks, topological measures of complexity, geometric optimization, Herlihy and Shavit's application of topology to asynchronous computation, information geometry, topoi in categorial logic, vector logic, homotopy type theory and ∞-groupoid structure of types, topological data analysis, persistent homology, quantitative homotopy theory, disconnectivity graphs, Morse theory in neural networks, Conley index theory in noise detection, manifold learning (dimensionality reduction) techniques like projection pursuit and Isomap, hierarchical clustering and any number of clustering algorithms on a metric space, topological clustering methods like Mapper, "partial clustering", functorial clustering, kernel PCA, one-inclusion graphs for concept classes, cubical complexes and hyperplane arrangements in sample compression, restricted Boltzmann machines and the renormalization group, herding and discrepancy theory, a variety of optimization scenarios like gradient descent or convex minimization, SVM, any variety of kernel methods, graphical models, k-nearest-neighbor search, random matrices in computational neuroscience, o-minimal theory and real algebraic geometry, sheaf theory for contextuality [1]. Given the interconnected nature of geometry and algebra, I also include some applications of abstract algebra to AI: Izbicki's algebraic learning models or "homomorphic learning", computable algebra and the theory of computable numberings, algebraic statistics.

Additional examples from category theory, mostly hypothetical: Lawvere's category of probabilistic mappings, category theory for systems engineering (e.g. a forthcoming category of design problems [108], the resource categories being developed at NIST [15], a hypothetical category for change propagation), category theory for reactive architectures in robotics including the hypothetical category of subsumption diagrams and the hypothetical category of "tuned" oscillators [58], McCullagh's category of design objects and statistical models [76], category theory for dynamical systems (e.g. Baez's work on chemical reaction networks, Spivak's work on dynamical systems and sheaves, Spivak's operad of wiring diagrams, the algebra of open dynamical systems [113], Fong's PROP of linear time-invariant dynamical systems), string diagrams in topological quantum field theory.

Broadly speaking, the above examples fall into three categories: applications of geometry to naturally spatial problems in AI, geometric studies of logic and computation, techniques of data analysis and learning defined on metric spaces, and categorical representations of all the above.

Suggestions for more examples would be greatly appreciated.

As of yet, there is no big Curry-Howard style correspondence between the two subjects. Homotopy type theory is probably the closest (types are ∞-groupoids where the given identity types are the path spaces), and there is something very interesting about Voevodsky's proposed univalence axiom as an "abstraction principle".[1] In a similar vein, Baez has described a (not quite formal) analogy between programs and cobordisms in the context of compact symmetric monoidal categories.

1.4 Examples of questions related to AI

The subject of "geometric intelligence" doesn't exist yet, much less does it have established questions and conjectures. In lieu of these we mention some broad problems in AI, with a bias toward ones with geometric and categorical applications.

1. The big one: statistical predicate invention / hidden variable discovery [29]. How do people do it? This is the technical formulation of what we call "the problem of learning structure", or concept formation, or, even more plainly, how to explain things.

2. Synthesizing knowledge, including learned knowledge, from multiple domains.

3. Many other more technical questions in machine learning. Discrepancy theory and the Gaussian correlation conjecture [18]. The sample compression conjecture.

4. Deep questions about models of computation. These include questions posed by Penrose and Smale (what are the limits of intelligence? Is artificial intelligence even possible?) which retract to questions about what we mean by "algorithm" or "computational process".

5. Questions posed by computational complexity theory. While directly related to AI vis-á-vis search and machine learning, it is also the quantitative reflection of "qualitative" questions about computability and models of computation.

6. Questions (really, entire fields of study) about how to implement human-like faculties in artificial agents. Computer vision, optical character recognition, natural language understanding, voice recognition, robot kinematics, motion and manipulation.

[1] By abstraction principle I mean something very concrete: if a description of a construct (e.g. a programming method) shows up in multiple places, you should be able to abstract it so that the description only shows up in one place. Also, see Section **??**.

Of course geometry and topology have their own problems and conjectures, and we will review a few in Sections 4 and G.5. To study "organizing principles" in any field requires understanding the relationship between a field's problems and the data of its examples.

1.5 A question

Question 1. Can we use cohomology to formalize assumptions about "structure in data" and thus about learning?

1.6 Reasons for studying cohomology as an AI researcher

Modern geometry is inextricably bound with this mysterious notion called cohomology. Mysterious because of its abstraction and because how there are so many cohomology theories in geometry and in topology, each useful in different ways. This is surely an overstatement, but it seems that for every subject or class of spaces there is a cohomology theory: simplicial homology for triangulated spaces, de Rham for smooth manifolds, étale for schemes, Floer for symplectic, (topological) K-theory for fibrations, sheaves for abstract varieties, l-adic for number theory, group cohomology, Hochschild for algebras, persistent homology for finite metric spaces, nonabelian for higher categories, and so on.

As an AI researcher, trying to understand cohomology theories in general seems like a useful thing to do for the following reasons:

1. Persistent homology is an active, important research program that explicitly uses homology in clustering and classification.

2. Topology and geometry, in various guises, have been used in machine learning since the inception of the later discipline, but rarely "qualitatively", e.g. using tools like homotopy and homology.

3. Sheaf (co)homology has met with some success in robotics [**?**] and in quantum computation [2].

4. Because cohomology is deeply connected to other areas of geometry and topology, a good understanding of cohomology theories in general should lead to an understanding of—or at least a new perspective on—other organizing principles in geometry.

5. Intuitively, learning is about extracting structure from data, and homological functors are ways of extracting the algebraic structure from a space. Whether or not this is a productive analogy we will see.

6. Even without making any progress on the hypothesis, writing all this down should help future students apply a wide range of geometric techniques to AI and learning. There is no textbook doing this currently, as far as I'm aware, though there are a number of related books from differential geometry and information geometry, e.g. Watanabe [118].

> "Typically, AI "succeeds" by defining the parts of the problem that are unsolved as not AI. The principal mechanism for this partitioning is abstraction. Its application is usually considered part of good science, not, as it is in fact used in AI, as a mechanism for self-delusion. In AI, abstraction is usually used to factor out all aspects of perception and motor skills. I argue below that these are the hard problems solved by intelligent systems, and further that the shape of solutions to these problems constrains greatly the correct solutions of the small pieces of intelligence which remain." - Brooks

2 Very brief review of AI

AI is a 60-year-old research program defined by four main approaches: symbolic methods, statistical inference, connectionism, and situated cognition. Broadly, the common goal is to design and engineer robots and virtual agents that can perform in a wide range of dynamic, persistent environments. To be clear, each approach constitutes a broad class of specific, computational *methods* for specifying behavior. I will not attempt a formal definition of "method"; the point is that the particular methods exist, can be implemented and recorded, and thus constitute a form of data about behaviors in the world.

Despite the title, this section is not quite a review. Our goal is to set the stage for a *structural description* of AI methods including but limited to the four approaches listed above, so that we can begin to think consistently and mathematically about their differences. We want to pass easily from one method to another by means of some overarching structure, carrying insights and structures from one setting to another without effort, "for free".

2.1 Approach: symbolic methods

Early progress in AI (from the 50s to 60s) focused on artificial agents directly capable of "reasoning", where the model for reasoning was mathematical logic. This bears itself out in three respects: as Moore [81] points out, not only was logic used to analyze the consistency properties of other reasoning systems, but it stood out as a model for such reasoning systems (e.g. Newell and Simon's Logic Theorist) as well as a programming language in which to implement such reasoning systems (e.g. LISP). Examples of this approach included knowledge databases like Cyc, robots like Shakey and Julia, and expert systems like Mycin and Dendral.

The underlying strategy of symbolic AI or logicist AI—as first set in [73]—is to extend logical methods to *commonsense reasoning*.[2] In essence, the idea is to compute *directly* on knowledge, for which purpose knowledge must first be abstracted into propositions. A specific system for abstracting knowledge into propositions was called a *knowledge representation* (KR); a variety of logics

[2] Compare this to its intellectual precursor: "the dominant goal, then, of philosophical logic is the extension of logical methods to nonmathematical reasoning domains." [111]

were then developed (often hand-in-hand with new KRs) to compute on sets of propositions: Robinson's resolution method, nonmonotonic logic and circumscription, default logic, description logics, modal logic and epistemic logic, inductive logic programming, tense/temporal logics, and a variety of calculi for reasoning about action. Most of these logics are ways of axiomatizing structural knowledge about the world—rules about time, place, causation, etc.—into the implication procedure so that deductions can be made computationally feasible.

Unfortunately, the first step of the symbolic strategy—making commonsense knowledge explicit—proved far harder than anticipated. Further, implementation of the strategy has been largely piecemeal; researchers tend to construct separate axiom systems that work on different toy problems without attention to how these axiom systems might interact on even a medium-sized problem [21] (much less on a real-world problem like motion planning). Worse, combinatorial explosion is still a huge problem, as it is in the larger field of automated theorem proving.

Question 2. Is there a paper somewhere that examines the interactions between (relatively large) axiom systems? This isn't quite the same thing as ontology integration, on which I have seen some papers [86, 102], though I imagine the two are related.

[TO ADD? J. McCarthy's description of circumscription, in relation to "in" and "out" in causal models.]

[TO ADD? Description logics and their use in data integration / validation. In answer to the question, where has KR been *really* successful?—well, in the design of databases and programming languages.]

2.2 The extension to knowledge representation, part I

While symbolic methods have fallen out of favor today, the field of knowledge representation is the right extension of historical work in symbolic methods, even though KR is in many ways a far wider and more broadly applicable field of study. Just as a symbolic method is a way of axiomatizing structural knowledge about the world into the implication procedure, knowledge representations are ways of imbedding structural knowledge about a given context into a possibly sub-symbolic method—or rather, from the perspective of this paper, the representations are ways of illustrating the structural assumptions already present in the method. Davis et. al. [22] describes five roles of KR, of which we cite two:

1. A knowledge representation is a *coarse theory of intelligent reasoning* expressed in terms of "(1) the representation's fundamental conception of intelligent reasoning, (2) the set of inferences that the representation sanctions, and (3) the set of inferences that it recommends".

2. A *medium for pragmatically efficient computation*, in the sense that a KR should organize information to facilitate making the recommended inferences.

Thus KR is concerned with the formalisms for representing knowledge[3], whether these be collections of propositions in a Frame language, semantic nets for words in a natural language, clusters in a vector space, concept classes $C \subset \{0,1\}^X$ in machine learning, or a simple table in a relational database (e.g. truth tables, lexicons, web ontologies, etc.). It is our belief, though we can offer no formal proof, that KR has applications to every approach in AI, not just the symbolic, since one must go with *some* form of KR in any practical AI application, even if the particular representation is implicit (as in an embodied robot) or opaque (as in the internal representations of a neural network). Wherever there is a consistent response to a challenge in the world, there is KR; this is the reasoning behind our very first claim: that AI needs "a systematic way of putting together logic with real-world data, and that statistical methods comprise only one class of options."

We reject the first two roles of KR described by [22]:

1. "A knowledge representation is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it."

2. "It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?"

These roles reprise KR's traditional emphasis on *abstraction* and *ontology*. But we believe that representation is involved in acting as much as it is involved in "thinking", and that one can have KRs where the set of ontological commitments is unclear or inaccessible, or where even the notion of ontological commitment is hard to define. Defining KR by these roles tends to obstruct a more structural understanding of KR. [Because...?]

KR, considered as a sub-field of computer science, is a strict subset of the symbolic approach; after all, representing knowledge in a computer requires, *a priori*, that the representation be formal, symbolic, and thus amenable to symbolic methods—everything reduces to a collection of propositions in some logic at some level. However, the choice of how to represent a (computational) problem matters in ways that are not obviously logical: it changes the computational complexity of the problem; it changes the available structure (think shortcuts) for resolving the problem; it changes our language for expressing the problem; it changes our intuition of how to solve the problem; it is, more often than not, *useful*. Much as the choice of number system can extend or constrain the possible solutions to a set of equations, with sometimes surprising and often informative outcomes, different KRs of the same environment can suggest radically, surprisingly different behaviors. [Find a good citation.] For example,

[3] "Knowledge" is a vague word. Perhaps it is objective and propositional; perhaps it is dispositional, situational, and embodied [30]; perhaps it is linguistic in some fashion that is neither strictly propositional nor dispositional but combinatorial and compositional [20]. These discussions are prior, speculative, "logical" in the sense of Question 3. [Justify this claim, expand into separate section?]

the choice to interpret a Bayesian network as a network of causal relations, as opposed to merely a list of conditional probabilities, forms the basis for practical and significant algebraic constructions (the do-calculus, [89]), even though a Bayesian network is, technically, just a list of conditional probabilities. Choosing or constructing the right representation is a difficult problem for any given context, one that resists easy formalization. It is the problem we hope to solve.

For a more classical discussion of KR, consider the papers by Newell [84] and Davis et. al. [22].

Question 3. How should we study and formalize the concept of a knowledge representation? This is the major question of this essay.

[Add more discussion of why this is hard, and a literature review (4 paragraphs?) of more modern approaches to defining KR.] Building on ideas coming from computational learning theory, Dreyfus [30], and the "Logical Consequence" seminar of Griffiths and Paseau [47], we give the following definition of KR:

Definition 2.1. A *knowledge representation*, or just *representation*, is a correspondence from a domain, not necessarily logical or formal, to a formal logic.

What is a domain, and what is a correspondence? Intuitively, when I say domain, I think of a domain of data, and when I say correspondence, I think of a correspondence in the sense of a set of constraints that bound the behavior of "learning algorithms" whose output (from mere propositions to more complicated mathematical constructions) can be analyzed via the formal logic. There are other possible definitions, fit for different questions. The definition above is sufficiently broad to cover all examples of interest in this essay, from the modeling of "non-logical constants" of Question 4 below to the typical dynamical system models of robotics all the way to the concept classes of Question 21. It grounds our later discussion in Section 6.

Question 4. In philosophy and especially analytic philosophy, one common domain is natural language. Another common domain is "common sense". If a KR is defined as a correspondence from some domain in which knowledge is possible to some formal logic, then we can analyze the success of a KR in several different ways. From one perspective, a KR is successful insofar as the logical notion of consequence captures the notion of consequence within the original domain. *But what is logical consequence, as opposed to "natural" consequence in the original domain?* For example, many philosophers [**?**, 47] hold that logical consequence is defined by its invariance over all non-logical constants.

[Why not put the "term paper" for Logical Consequence here?]

Naively, one could interpret a KR as a mapping

$$\begin{tikzcd}\text{Domain} \arrow[r, "R"] & \text{Logic}\end{tikzcd}$$

$$\begin{tikzcd}
s_1 \arrow[r, mapsto] \arrow[d, "\Rightarrow"'] & p_1 \arrow[d, "\vdash"] \\
s_2 \arrow[r, mapsto] & p_2
\end{tikzcd}$$

[What about generalizing the correspondence from domain to formal logic, so that "logic" is not some abstraction but is "experienced" inside a computer? Or logic may live in the physical world as physical laws. This should be a key part of how to think about *simulations*.]

When the domain itself is formal, it makes sense that the representation can also be specified formally and exactly. But what if the domain is not formal? At least the representation should be "consistent" in some sense; that is, it should be at least possible to find evidence related to the domain—*data*—for whether the behaviors predicted by the formal logic correspond to the domain, even if there is no global, formal rule that guarantees that all the data (or any of it) makes sense. So even if we cannot reason formally about KRs when the domain is not logical, we can still think *behaviorally*: that is, interpret each KR as a more-or-less formal way of delimiting or inferring a "space of possible behaviors" in a domain, and the domain itself can be thought of as the full space of possible behaviors (this may not always be reasonable). Each such domain is endowed with various structures and descriptions: e.g. a mathematics classroom, a manifold in $\mathbb{R}^4$, a fragment of the English language, a "concept space" of right and wrong hypotheses, and so on.

Though we should question what makes it possible to think behaviorally. Again, the domain may or may not be formal, while "logic" is, by assumption, formal. Recall Moore's point: formal logic is both a method for studying informal domains and a domain itself amenable to study. By analyzing the space of possible behaviors and inferences, we can think of each representation as a way of passing from a consistent, perhaps formal system of study to more fundamental stories about structure. To say that it is possible to study a representation behaviorally is to say that each representation is not only as a correspondence between a domain and a logic but an example of a correspondence *from logic to geometry*. [Geometry has to do with data integration, though I'm not sure a representations go all the way to data—maybe we should call such a thing, something just prior to a true geometry, a "geometry up to simulation".]

Perhaps we can borrow some ideas from the foundations. From Rodin's remarks [**?**], work in topos theory has shown that the semantics/practice of a rich structural theory (e.g. Euclidean geometry) cannot be controlled by its axiomatization (such as in the *Elements* or in ZFC) but requires additional notions of basic elements like points and lines and the rules for putting them together. (Surely this was obvious before, but our intuition for what was missing from the axioms was formalized through the (∞-)topos structure.) Further, these rules and the knowledge of how to use them *cannot* be reduced to mere propositions; the difference between constructing the line through two points (as happens in a proof) and the postulate "there exists a line through every two points" is not trivial. This feature was once observed by Hilbert in the distinction between "genetic" mathematics (by which Dedekind constructed $\mathbb{R}$ from a primitive $\mathbb{N}$) and his own formal program. [**?**] suggests that this "how" knowledge may be encoded in a dependent type theory, or in a homotopy-theoretic setting via homotopy type theory. If so, then could each KR be a sort of abstraction principle for (comparing and combining) types? See Section 6.

Question 5. How do advances in logic relate to advances in KRs connected to that logic? For example, how can theorems about propositional logic be lifted to theorems about all possible representations that outputs into propositional logic? How can theorems about dynamical systems be lifted to theorems about all possible representations that output into a dynamical system? [Discuss in Section **??** on a copresheaf representation? This question puts me in mind of another question, about the relationship between inductive inference and its "polynomial-obsessed" cousin, computational learning theory.]

Question 6. Spivak [102] describes "ologs" as a categorical knowledge representation. An olog is, roughly, a way of replacing a relational database with a category, with advantages accrued from comparing and combining databases (think ontology integration / ontology matching). In what ways can this address some of the known historical issues with the KR-as-ontology approach, and in what ways does it fall short?

2.3 Approach: statistical inference

Inductive inference is the process of hypothesizing general rules from particular examples; the essential difference from deduction is that 'correct' induction may nonetheless give rise to false conclusions: there is a fundamental uncertainty in induction.[4] Statistical inference is inductive inference on examples drawn from a probability distribution, where the distribution quantifies the uncertainty. Examples of this approach include neural networks, hidden Markov models, logistic regression, and any number of other machine learning algorithms (though it does not include inductive learners like FOIL, where inference is over propositions).[5]

Think about what it means to "draw from a distribution". I hesitate to call it a strategy, but there is a common assumption in every statistical algorithm which is built into the very idea of sampling from a distribution: all examples are equated under a *statistical model*. In practice, every statistical model requires (though does not fully specify) a process for regularizing its examples, which often comes down to killing any relations that might exist between the examples.

[4] Gold's seminal paper [43] introduced the standard recursion-theoretic formalism for learning: given a sequence of examples $(f(0), f(1), ..., f(n))$ of a computable function f, is there a learner that can produce f? [Take out Wikipedia's definition and put in Angluin's.]

[5] Statistical inference is indelibly associated with the word *learning*, though it's useful to point out that not all learning involves inference, much less statistical inference. For example, a machine learner can learn a new rule by *being told* the rule by either a programmer or another machine. Alternately, it can learn the rule by applying a known analogy with another, known rule. [25]

(We might think of this regularization process as type-checking the data, or as truncating the higher path structure in the sample space.) The output of this process is what we call "data", whether in the form of vectors, lists, or bitstrings. I think of regularization as an intrinsic part of working with data, and of the experimental process more generally.

We can look at this in two ways. (1) Take the regularization process as evidence that each statistical model, at its assembly level, commits to a single representation of the world.[6] (2) Alternately, take the statistical model as evidence for various properties of the representation. More baldly: take the statistical model as a set of constraints on "good" representations of the data. These perspectives are important when considering how to vary our representations over different models, or how to vary our models over different representations—over different sorts of data.

Some examples. Sometimes regularization is obvious; after all, a class label is a kind of relation, and the entire point of classification is to recover these labels. In other cases this process is not so obvious, as when scientists preprocess data in order to remove (structured) noise incident to the measurement apparatus. For example, in quantum mechanics one has the idea of "preparing a state". The state represents a system for which we do not know (or care) about where it came from (i.e. how it was produced); just that we we have the output of that system and that it is in a particular state. We do not care about the previous history of the state.

I think of "preparing a state" as a generalization of the process of regularizing a statistical unit, since any number of physical and mathematical procedures (mirrors, lasers, beam splitters) may be involved in preparing a quantum state, while we are focusing on a small subset of mathematical procedures in the case of statistical regularization. All the relevant structure in a statistical unit is taken to be *internal*: for example, a vector is represented completely by its ordered list of components. In this view, such statistical models are the tools by which we equate data and 'prep' it for induction; perhaps another way of stating this is that data, unlike the variables provided in a logical domain, are always "quantified"—in the sense of existential or universal quantification—by the representation which introduced them.[7]

[Also: discuss the possibility and requirements for a set of E-M axioms for statistics and causation. Cf. McLarty's paper for inspiration?]

As one might expect, simplifying data by killing relations allows one to apply a uniform inference procedure that works well on large data sets. Practically, this has worked well since technology tends to produce such large, simplified data sets. Several challenges arise though:

1. The curse of dimensionality. Whether in classification or in clustering, high-dimensional data can vastly increase (depending on the choice of

[6] One can always extend the representation to enable higher relations (though not easily), but one cannot get away from committing to a representation.

[7] Perhaps another way of stating this: the model understands what a vector (or table, or graph) is and is capable of "killing" (or unwrapping?) the external vector structure in order to compare and differentiate the data inside.

learning algorithm) the number of examples required to demonstrate a significant result.

2. Collecting, cleaning, and integrating data is expensive and dismayingly ad hoc. While we no longer have to hand-code behavior, the flip side is that we are now doubly dependent on having good data, especially labelled data.

3. The design and choice of reward functions (or loss functions) can be very difficult and can obscure the contribution of the programmer versus the actual "intelligence" of the learner.

4. It's hard to learn structure. Statistical learning tends to be "shallow": we can learn basic concepts (the difference between plants and animals) but find it harder to learn the kind of complicated relationships between concepts which are typical in logic and higher reasoning.

Question 7. How do symbolic and statistical approaches differ on their approach to recursion?

Consider: a general strategy in machine learning involves *online* learning, where one forms a hypothesis then updates that hypothesis by testing it on single points of data (or possibly small batches). There is an incremental, recursive nature to such online methods, and the recursion is particularly simple: just update the hypothesis. Would this recursive property fail if we tried to apply online methods on data endowed with higher structure, for example grammatical structure? I.e., what would happen if our algorithm ate not a vector of words, but sentences with an implicit notion of word location and colocation? (One imagines not just a vector of words but a "chain" of words, in the sense of a chain complex, or an open set via a sheaf.) Certainly the problem is harder, but in what way is it harder; can we quantify that relationship? If statistical quantification ("truncation"?) is thought of as an operation on individual points of data, how could it interact with the recursive operation "update model on data"? [How could I state this question be better stated? Answer: better examples.]

I guess we could always "preprocess" the more complicated data so that we can feed it in a form accepted by a typical statistical algorithm, but for the results to be useful, this preprocessing step requires a careful understanding of the data and where it comes from.

But does the question "where does the input come from" matter in traditional recursion theory?

What is a statistical model?

To do statistical inference at all we must use a statistical model to connect probability distributions to samples; these distributions formalize further assumptions about our random variables X and "where they come from". The sum of all these connections can have a surprisingly complicated logic. We briefly review the categorical formalism suggested by McCullagh [76]. The essential idea is that the meaning of a model should be preserved (via a functor) under all the usual changes (morphisms) in the sample, the covariate space, or the labels: "the sense of a model and the meaning of a parameter, whatever they may be, must not be affected by accidental or capricious choices such as sample size or experimental design."

Definition 2.2. Let $\mathcal{U}$ denote a set of *statistical units* (e.g. trials in an experiment). Let $\mathcal{V}$ denote a *response scale* (i.e. a set of labels). Let Ω denote a covariate space (i.e. a feature space). Then a *statistical model* or *model object* is a map

$$P : \Theta \to \mathcal{P}(\mathcal{S})$$

from a parameter set Θ (defined for some covariate space Ω) to the set of all probability distributions on the sample space $\mathcal{S} := \mathcal{V}^{\mathcal{U}}$.

Now the idea is to associate to each *design object* $\psi : \mathcal{U} \to \Omega$ a statistical model $P_\psi : \Theta_\Omega \to \mathcal{P}(\mathcal{S})$ in a way that respects all the 'usual' changes in $\mathcal{U}$, Ω, $\mathcal{V}$, and the parameter set Θ_Ω. We do this by defining $P : \Theta \to \mathcal{P}(\mathcal{S})$ as a natural transformation of functors. The point of this is to impose a consistency condition on computing probabilities [a coherence condition up to (higher) homotopy?], so that "different ways of computing the probability of equivalent events give the same answer" [76, pg. 1241].

Question 8. Often we do well enough without describing statistical models explicitly (witness algorithms like SVM or AdaBoost that never even mention probability), and in most scientific applications a good statistician hardly needs to refer to the abstract nonsense just discussed. However, McCullagh's formalism could be useful for clarifying and analyzing certain tacit assumptions that go into the inference procedure. Particularly, can we use it explore generalizations of the "design object" $\psi : \mathcal{U} \to \Omega$, i.e. designs that do not truncate the "higher path structure" that may exist in $\mathcal{U}$ but lift them fiber-wise into Ω?

2.4 Approach: connectionism

[To add: discuss Smolensky's ICS, the role of hybrid architectures even in connectionism, and its applications in NLP. Add discussion of how every matrix W of composed connections + weights encodes a kind of implicit causal order (do you interpret "clowns tell funny jokes" as clowns + jokes first, or tell, then clown + jokes, then funny?). Do you first focus on the object or the subject?]

To add: from the perspective of graph theory / network science, "topology" is represented by the unweighted adjacency matrix, while "geometry" is represented by the weighted adjacency matrix. It kind of makes you realize that geometry in the sense of a metric is really assuming a lot, and also that any kind of weighted network defines the contours of a certain geometric object.

Within the field of artificial intelligence, connectionism is the idea that intelligent behavior arises from the emergent properties of networks of simple automata. For complicated reasons, connectionism today has retracted largely to the domain of statistical inference.[8] AI researchers do not claim that artificial neural networks can simulate human intelligence on any level except on restricted, well-defined learning problems—though in such cases (classification, natural language processing, etc.) we have lately seen outstanding results and a resurgence of neural network methods under the moniker "deep learning". Today, connectionism in AI is synonymous with McClleland & Rumelhart's paralleled distributed processing (PDP) approach [75], which emphasizes the massively parallel architecture of the brain along with the distributed representations of concepts and relations, which are stored as weighted connections between many simple neurons. See [74] for a more recent note.

There are two main problems with connectionism in AI: first there is the obvious one of learning structure inherited from statistical inference. The second problem, one common in AI, lies in the vast divide between a very plausible but general proposal about human cognition and the engineering task of constructing an artificial model at anywhere near the requisite scale and complexity which exists in the human case.[9] Nor is there any obvious research path today that will take us through the no-man's-land between the general proposal and our toy models of neural networks. Illustrating the problem, von Neumann writes (in a 1946 letter to Norbert Weiner):

> "What seems worth emphasizing to me is, however, that after the great positive contribution of Turing-cum-Pitts-and-McCulloch is assimilated, the situation is rather worse than before. Indeed, these authors have demonstrated in absolute and hopeless generality that anything and everything Brouwerian [i.e. constructible, "computable"] can be done by an appropriate mechanism, and specifically by a neural mechanism—and that even one, definite mechanism can be 'universal'. Inverting the argument: Nothing that we may know or learn about the functioning of the organism can give, without 'microscopic,' cytological work, any clues regarding the further details of the neural mechanism... I think you will feel with me the type of frustration that I am trying to express."

In neuroscience

As a larger trend in cognitive science, connectionism has the most cachet in hard neuroscience, where some variant of it is unspoken but nearly universally assumed by the working neuroscientist. The search for a general theory of "brain intelligence" splits along three approaches:

[8] Though it's of some interest that McCulloch and Pitts, who first originated connectionism as a theory of intelligence, viewed neural networks as closer to logical methods, akin to universal Turing machines, than as methods of statistical inference.

[9] The brain is complicated on both a local scale (consider specialized areas for processing vision, language, spatial navigation, fine motor control) and a global scale (connections between distant brain regions, the large-scale structure that undergirds the ability integrate percepts, models, and behavior). In practice, artificial neural networks are usually simple on both local and global scales.

1. the computational modeling of large scale brain dynamics, seeking patterns of synaptic potentials over a large network ($n \geq 6 \cdot 10^9$). "How do networks of neurons give rise to complex behavior and learning? What can statistical mechanics say about brain function?"

2. studies of particular regions of the brain associated with cognitive faculties (e.g. vision, olfaction, and/or various pathologies arising from local lesions). "How does the brain see? How does it navigate in space? How does the brain form memories?"

3. cognitive neuroscience (along with cognitive science) blends biological theories and biological data from MRIs and encephalograms with psychology-style experiment design. "What does the brain look like when it feels pain? How do activations for different stimuli compare?"

There are a variety of theoretical viewpoints that try to synthesize some of the above data: Walter Freeman has a general theory that applies dynamical systems theory to brain function. Ehresmann and Vanbremeersch [34] have a model they call "memory evolutive systems" that uses category theory to model neural behavior (there's an interesting idea of describing colimits as models of collective behavior). Llinás [66] describes a embodied cognition framework where subjective "qualia" are the subjective correlates to the physical brain mechanisms that bind together multiple sensory representations. From a more philosophical perpsective, Tononi [**?**] has an interesting theory that measures the consciousness of some artifact (e.g. a brain region) based on its capacity to integrate information.

Excepting Tononi's theory, these are all theories of *human* intelligence, and while we can take inspiration from them, we should be guarded about how applicable they are to designing a functioning artificial intelligence. Birds may fly by flapping, but planes do not, and rockets even less.

2.5 Approach: situated cognition

In AI, situated cognition is often associated with robotics, though it has antecedents in early (non-robotic) examples like ELIZA and SHRDLU that emphasized "scruffy" programming solutions to real-world problems over "neat" mathematical approaches involving either logic or statistics. The emphasis of situated cognition was and still is on getting interesting behavior in a *particular situation* (which may resist mathematical representation) rather than on obtaining formal justification for a method; such situations arise dynamically through the interaction between an environment and the body of the AI. See [30] for a philosophical account of this view.[10] In Brooks' foundational paper [17], situated cognition comes across as an engineering philosophy: "to use the world as its own model". Practically, this means engineering simple systems in the real

[10] For some more discussions in philosophy of mind with applications to AI (esp. on "intentionality", the capacity of minds to form and give meaning to mental representations), see Dennett [24], Fodor [37], and Putnam [91].

world (as opposed to complicated algorithms on cleaned data) and then incrementally improving these systems in order to obtain more complicated behavior. Examples include Grey Walter's tortoise, Braitenberg's "vehicles" [14], Brooks' Genghis and the subsumption architecture [17], a series of self-driving cars from CMU and from Google, and a variety of biologically-inspired approaches.[11]

One might expect that work on the particular, localized problems (particular algorithms in learning, particular robots in particular environments) would be orthogonal to understanding higher cognition. Situated cognition suggests that two may not be as separate as we once thought. That is, higher cognitive abilities like language learning and memory formation may not be separate, "universal" processes but arise as elaborations and conditions on pre-existing a family of behaviors, all of which were adapted to specific environmental contingencies.

The focus on engineering in situated cognition is critical. Brooks' subsumption architecture (see Figure 2.1) is an engineering architecture: a series of guidelines for combining and composing multiple simple, simultaneous behaviors in a partial ordering of "layers", with the condition that lower-layer behaviors have precedence. So in principle it uses a low-level KR, but the KR was not what was important to subsumption. The subsumption architecture managed to produce impressive results in locomotion because (1) it was specifically engineered for locomotion, which really comes down to the fact that (2) it was *tightly integrated* with simple sensors and motors in the world.

Unfortunately, designing purely reactive "bodies" along with an appropriate architecture (as in subsumption) is a time-intensive task, much like hand-coding rules in an ontology.[12] *Hybrid architectures* exist that marry high-level KRs with

[11] There is a much broader literature on situated cognition in philosophy, psychology, and cognitive science which we can reference only in passing; the body's role in cognition (compare to the brain's role in locomotion) has a rich intellectual history going back through Heidegger to Kant to medieval theories of the four humours. For example, the thesis of *embodied cognition* claims that higher cognitive abilities like abstraction and language are determined by (or supervene on) lower-order behaviors such as sensing and moving. This thesis is often paired with the thesis of *embedded cognition*: cognition depends on the natural (and social) environment—the richness and 'intelligence' of human behavior cannot be explained solely by the complexity of the brain but depends largely on the richness of the world in which that person lives. Perhaps these claims are true—certainly human cognition tends to be very well integrated with our sensorimotor skills (a baby raises its finger, points at an object in the distance, and says "cat"). But true or not, we can build something interesting on the idea.

[12] In some ways this present essay is motivated directly in opposition to a quote from Brooks [17]: "An extremist might say that we really do have representations but that they are just implicit. With an appropriate mapping of the complete system and its state to another domain, we could define a representation that these numbers and topological connections between processes somehow encode." In fact I *am* an extremist and I will call these representations implicit. By representation here Brooks means a typical logical representation; he is right that there is little point to constructing such a traditional, propositional semantics for his low-level "program" of topological connections—one feels that this would actually be taking a step back if our goal is a structural comparison of methods, since invariably the logical representation would be more complicated than diagrams such as Figure 2.1, we lose the computational details of the circuit design, and in any case we lack good tools and theorems for analyzing, much less combining, these sorts of axiom systems (cf. Question 2). But first, it may be interesting to think about "appropriate" (i.e. functorial?) mappings to other, non-propositional domains. And second, the answer is not to avoid thinking about representations, or to use

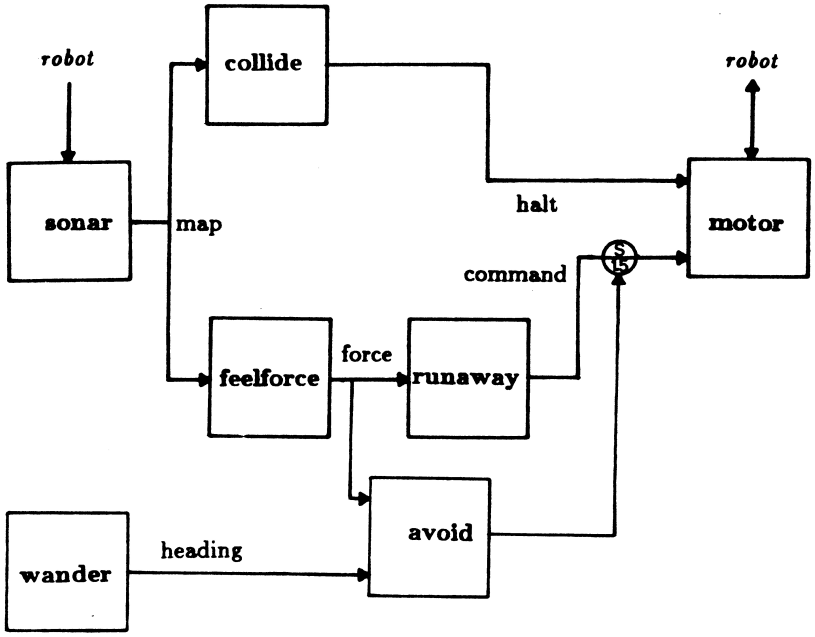


Figure 2.1: A relatively simple subsumption diagram, taken from [16].

statistical inference (e.g. on sensor data, or to calibrate certain defined behaviors) over a base of reactive behaviors tightly integrated with sensors and motors in the real world. Examples and acronyms abound: SOAR, CAPS, Copycat, PRS, ROS, etc. Alternately, consider physical examples like the Google Car, the new Baxter robot, or even the lowly Roomba. Such architectures are among the state-of-the-art in AI currently, and they emphasize the need for meta-level control of many different sensor modules, reasoning modules, and many, many reactive motor modules; one (vaguely connectionist) solution has been to organize these modules into layers that communicate via control rules. As one can imagine, such architectures can become extraordinarily complex very quickly as their components multiply, and their inability to learn new representations has limited their application to problems like language learning.

Question 9. A hybrid engineering architecture is a syntax (sometimes a graphical syntax) for combining and composing methods (thought of as arrows). We can model this syntax through a variety of means, for example as wiring di-

only implicit representations—to hack. Hacking is necessary, but it depends on a foundation of tools, languages, and structures that must be progressively expanded and reformulated for different situations and environments—and this foundation is built from the bricks of knowledge representation. [This may not be quite right; it feels like changing the KR is not just about changing computational language or changing libraries, e.g. syntax, but also changing the terms by which we evaluate performance.]

agrams [101]. But in what way can we model something like "tight integration with sensors and effectors" in the category?—this is the abstract reflection of questions about "learning through interaction" considered by Oudeyer [88], Pfeifer [90], Wolpert [120], and many others. [This list really deserves its own appendix.] This question points to disciplines like reinforcement learning, which abstract this interaction between agent and environment (also called an action-perception loop) using reward functions. (See [**?**] for a nice information-theoretic extension to the reward function story.) It is also considered by Koditscheck [**?**] as a question about of coupling different control problems together, where tight integration with sensors and effectors may be thought of as a story about convergence (?) between the control problems. [It may make sense to talk about Haskell and FRP here: the distinction between "pure" and side effects, but somehow to live in the real world we need to manage the side effects.]

Recall that we are after a structural description of AI methods, so that we can think consistently about their differences. In my mind, a structural description of AI methods ought to be the abstract analog to an engineering architecture for AI, but to be useful this structural description, like the engineering architecture, must vary with respect to different assumptions about the nature of the world/task environment and the agent's ability to interact with it (cf. "body" in embodied cognition). With reference to Question 3, our definition of KR must either vary with respect to such assumptions or abstract from them in a consistent but *nontrivial* way. (It is hard to describe what we mean by nontrivial, though certainly we do not want to abstract completely from or 'zero-out' all external assumptions about the environment.) I hypothesize that this requirement is the major obstacle to extending the KR of logical methods to the KR of situated methods (and also, to some degree, the KR embedded in a statistical model), and thus to a structural comparison of all methods.

Question 10. This question is generalizes the question of sensor integration: suppose that, according to the hypothesis of embodied cognition, that people embody their emotion and cognition in objects and processes outside of their minds: in their physical bodies, in physical objects in the world, and even in abstract things like norms, institutions, and language. When people offload cognition into something outside of themselves, what do those objects and processes look like? Further, how should we talk about "trust" or "confidence" (as opposed to trustworthiness) as a measure of the degree to which each object or process is integrated into the persons own processes? This generalizes the question of sensor integration since sensor sources are just one class of embodied object or process.

This question originally arose in the context of urban studies: "how do we better integrate sensor data into the decision-making processes of communities and governments?"

My hypothesis: we can use the idea of embodied cognition to rigorously analyze the compositional properties of those emotional states and cognitive processes. I develop this further in this scratchpad.

From Brendan's talk. Factorization hides *internal structure*. [Can I connect this to both the weak factorization systems in homotopy, as well as with the definition of "internal structure" aka structure in data?] E.g. Epi-mono factorization systems as in FinSet. A "corelation" is a equivalence class of cospans "stable under pushout", which uses the factorization system in that category. I.e a corelation discards the internal structure of a cospan... it models black-box composition. Another way of thinking about this: cospans accumulate structure i.e. representations of the empty equivalence class; corelations forget this.

Related to factorization; a category that has corelations "as a pushout" / a factorization system should also satisfy the pullback-pushout property. f-¿¡-g = p¡–¿q if pq is a pullback of fg; alternately f¡–¿g = p-¿¡-q if pg is a pushout of fg.

[applicable to neural algebra? Controllability is defined in terms of being able to change trajectories. We can start on one trajectory and end on another trajectory. But there are certain signal-flow diagrams where "there is no trajectory that takes us from one traj to another". Theorem: an LTI system is controllable iff it has a span representation. This allows us to analyze the controllability of the larger system by analyzing local properties.]

"Complexes good, (co)homology bad." - Richard P. Thomas

3 Very brief review of algebraic topology

Recall our initial hypothesis: cohomology theories can be used to formalize "assumptions about structure in data". So, morally, each cohomology theory embodies an approach to structure. (Yes, we suffer from some pretty loose morals in this paper.) One-half of this discussion depends on a looser, more general definition of "data", which we defer to Section 6. The other half can be understood as the answer to the following question:

Question 11. Why analyze spaces at the level of cohomology at all as opposed to, for example, at the level of chain complexes? Why analyze spaces at the level of cohomology as opposed to at the level of zero-sets of polynomials? What, in principle, is different or impossible without turning to cohomology?

On the surface, the answer is that cohomology is at least somewhat computable, and thus its use is almost necessary in topological applications where we want answers rather than more theories. Of course this answer is pragmatic [wc: *a posteriori*?], and tells us nothing about why cohomology was invented in the first place or why we might expect it to be useful before knowing any applications. The full answer is implicit in the theory of derived categories and their descendants, where we take our "invariant" as the full (co)chain complex itself rather than its (co)homology. [The primitive idea is that the computation of the (co)homology was already inherent in the definition of the chains.]

In general, any answer to why cohomology matters will depend on the system—the "organizing principle", a yoga based on experience with actual proofs—we use to organize our invariants.[13]

We assume the reader is familiar with algebraic topology up to the level of, for example, Hatcher [50]. The goal of this section is to paint a picture of the development of cohomology in topology and to review and motivate the definition of the (homotopy) category of spectra, on the way to an abstraction of cohomology suitable for AI. There will be some major diversions into homological algebra. Most of the material and definitions presented here are taken from Hatcher [50], Lurie [69], notes from the 2002 motivic homotopy summer school [32], and, of course, the nLab. Since this section is to some degree historical, I have borrowed liberally from the surveys of May [71] and Weibel [119].

As usual, all maps are continuous unless stated otherwise.

Question 12. Can we characterize what we really mean when we say that (co)homology is computable? Perhaps a good starting point is the question, what *is* a spectral sequence, really? At least in the Serre case, the idea is that certain kinds of cohomology have this nice recursive structure where we can take the "cohomology of the cohomology", so that the cohomology on one page reduces (word choice?) to the cohomology of the next. Furthermore, this recursive structure makes the cohomology computable (in the sense of tractable or solvable, not necessarily formally computable), and is in fact probably one of the best formal explanations for why we think of cohomology as "easier" than homotopy. It is, further, curious that spectral sequences have a much more direct algorithmic, "machine-like" definition than most objects in topology. In some sense, they are assembly-level representations of the process "required" by a cohomology theory. Of course the recursive structure is natural since it all comes down to ways of talking about filtrations of spaces or other, more explicitly "derived" objects (the nLab, for example, talks about spectral sequences as obtained by filtering a stable homotopy type then applying a homological functor—somewhat crazily, it even makes sense to talk about the "abelian category of bigraded spectral sequences"). But still—are there other explanations for this recursive structure, are there deeper ways of thinking about it and thinking about spectral sequences? I'd guess that this will take some work really understanding the definition and structure of the differentials. The reason I ask this question is because many observations in stable homotopy seem to be encoded in (and read off of) various spectral sequences—fibrations to the Serre SS, higher cohomology operations to the Adams SS, cohomotopy and K-theory to the Atiyah-Hirzebruch SS, etc.—and it's not unreasonable to think that thinking about spectral sequences in a different way could help abstract and simplify a large number of other results in topology.

[13] One might ask, why is all that "cool Grothendieck stuff" so much more glamorous than solving polynomial equations? In a different direction, one might ask, why are linear approximations so useful?

3.1 Axiomatics

An ordinary (co)homology theory is a family of functors that satisfy some version of the Eilenberg-Steenrod axioms: functoriality, long exact sequences, homotopy invariance, excision / Mayer-Vietoris, and dimension. Specifically, these are:

1. Functoriality: $H^n : \mathbf{Top} \to R\text{-}\mathbf{Mod}$ is a functor.
2. LES: for any pair (X, A) there is a long exact sequence of the form
$$\cdots \leftarrow H^n(A) \leftarrow H^n(X) \leftarrow H^n(X, A) \overset{\delta}{\leftarrow} H^{n+1}(A) \leftarrow \cdots$$
3. Homotopy invariance: $X \overset{h}{\sim} Y$ implies that $H^n(X) = H^n(Y)$
4. Excision: if (X, A) is a pair and $\bar{U} \subset A^\circ \subset X$, then the inclusion map $i : (X - U, A - U) \to (X, A)$ induces an isomorphism in cohomology
5. Dimension: $H^n(*) = 0$ where $* = \{pt\}$

Of these, excision is probably the subtlest and most geometrically substantial, since it redounds to the observation (often presented via Mayer-Vietoris) that we can pass from a space X to its open covers through, for example, subdivision, and then try to re-stitch the cohomology $h(X)$ from its cover via a homotopy pushout.

Each cohomology theory may also come in a variety of flavors depending on the choice of coefficient group. There are other nice structures defined on cohomology: cup product, cohomological operations and Steenrod squares, bialgebra structure, classifying spaces, Postnikov towers. And a wand for computing: spectral sequences. Remove the dimension axiom and we obtain extraordinary cohomology theories like topological K-theory and complex cobordism.

The axioms serve as an *interface* between algebra and topology and in that sense were the culmination of over a century's work; as [71, pg. 3] points out, one of their great virtues is that they "clearly and unambiguously separated the algebra from the topology" as part of the developing separation of homological algebra from algebraic topology in the 1940's and 50's. Before that, these subjects shared a common prehistory in the classical study of "connectedness numbers" and homology numbers by Riemann, Betti, and Poincaré in the nineteenth century.[14] The axioms or something like them were probably inevitable as early as De Rham's theorem in 1931 (or even as early as Stoke's theorem). By solidifying our understanding of (co)homology, the axioms provide a basis for exploring

(1) other possible interfaces between algebra and topology, especially homotopy

[14] Actually the history of topology between Poincaré and Eilenberg is quite interesting, and it suggests some rather different emphases—particularly on duality and cohomotopy—than the standard, pre-spectra curriculum does today.

(2) similar, “cohomological” interfaces between other categories, cf. Section 2.3.

We are interested in (2). However, this later story depends crucially on an understanding of (1).

Question 13. Has there ever been a case where we constructed a new cohomology theory directly from the axioms in a formal way, by adding axioms?

Eventually it will be important to note the connections between spectral sequences and persistent homology [33], which might give us a direct line to the assertion “cohomology formalizes assumptions about structure in data”.

3.2 The category of spectra

We can readily see from examples that homology is a *stable* phenomenon, e.g.

$$h_n(X) \simeq h_{n+1}(\Sigma X)$$

where Σ is our usual (based) suspension functor. Less readily, we can see that homotopy is an *unstable* phenomenon, which may be understood as the negative of the statement above or, as in Blakers-Massey, as the failure of the excision axiom to hold in all dimensions. I will not explore this line of thought, which arose from Freudenthal’s famous observation of a “stable range” of dimensions for the homotopy groups of spheres, except to point out that spectra are one setting in which such theorems and observations—along with a raft of duality theorems (Poincaré, Alexander, Spanier-Whitehead)—become obvious. Spectra, in other words, are a convenient language for stable homotopy. Stable homotopy, of course, is important because stable homotopy *groups* (of maps) are the natural linear approximations to homotopy *sets* (of maps).

Often we want to study the space of (basepoint-preserving) maps from a topological space X to a topological space Y, and a reasonable structure on this hom-space is the set of homotopy classes of based maps, denoted $[X, Y] := \pi_0(\text{hom}_\bullet(X, Y))$. Unfortunately, the set $[X, Y]$ does not usually have any algebraic structure; only in certain special cases can we analyze $[X, Y]$ as a group, much less an abelian group (as we do in cohomology). It turns out that we can organize these special cases under the heading of spectra.

Suppose X is the suspension of another pointed space X'. Then $[X, Y] \simeq \pi_1(\text{hom}_\bullet(X', Y))$ admits a group structure by the definition of π_1. Further, if X' is itself the suspension of some space X'', then $[X, Y] \simeq \pi_2(\text{hom}_\bullet(X'', Y))$ will be abelian since π_2 maps to abelian groups (this fact is elementary for π_2, but for higher π_n it follows from the Eckmann-Hilton argument). One can extend this to natural maps

$$[X, Y] \to [\Sigma X, \Sigma Y] \to [\Sigma^2 X, \Sigma^2 Y] \to ...$$

where, intuitively, each $[\Sigma^k X, \Sigma^k Y]$ is a linear approximation to $[X, Y]$. Assuming X, Y are finite, pointed CW complexes, the Freudenthal suspension theorem

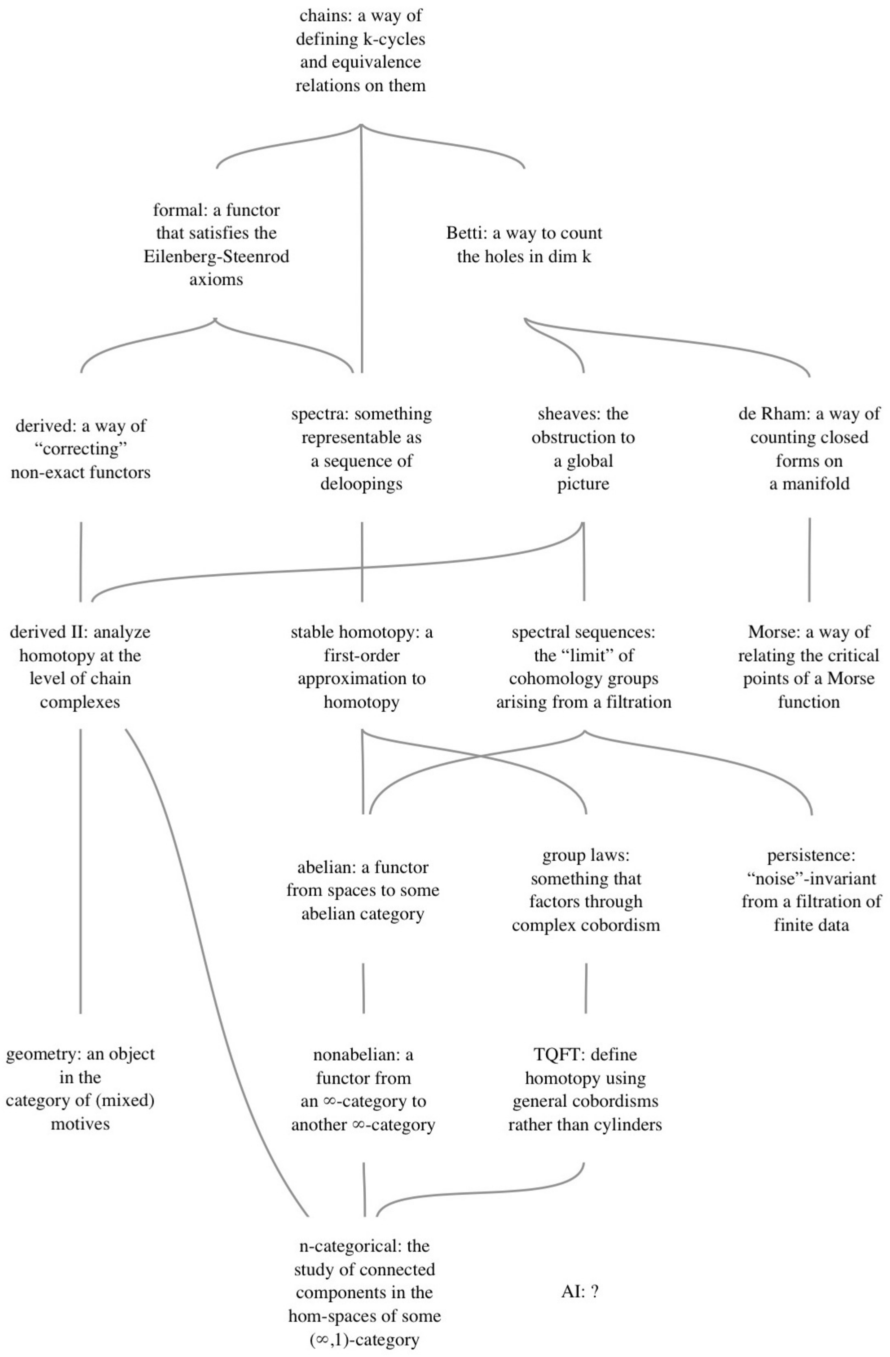

chains: a way of defining k-cycles and equivalence relations on them
formal: a functor that satisfies the Eilenberg-Steenrod axioms
Betti: a way to count the holes in dim k
derived: a way of "correcting" non-exact functors
spectra: something representable as a sequence of deloopings
sheaves: the obstruction to a global picture
de Rham: a way of counting closed forms on a manifold
derived II: analyze homotopy at the level of chain complexes
stable homotopy: a first-order approximation to homotopy
spectral sequences: the "limit" of cohomology groups arising from a filtration
Morse: a way of relating the critical points of a Morse function
abelian: a functor from spaces to some abelian category
group laws: something that factors through complex cobordism
persistence: "noise"-invariant from a filtration of finite data
geometry: an object in the category of (mixed) motives
nonabelian: a functor from an ∞-category to another ∞-category
TQFT: define homotopy using general cobordisms rather than cylinders
n-categorical: the study of connected components in the hom-spaces of some (∞,1)-category
AI: ?

Figure 3.1: Perspectives on cohomology

tells us that such a sequence will always stabilize, and we define the stable result as $[X, Y]_s := \text{colim } [\Sigma^k X, \Sigma^k Y]$. From the above we know that this is an abelian group, and we call it the *group of homotopy classes of stable maps from X to Y*.

So we are interested in these different approximations of $[X, Y]$ and how they relate to $[X, Y]$ and to each other. But instead of considering all these different homotopy classes of maps in different degrees, we can consider just one *map of spectra* which carries the essential information. The following definitions come from [32] (note the replacement of simplicial sets for topological spaces).

The homotopy groups π_n fall badly at being homology theories, in the sense of being preserved under pushouts and cofibrations (all of the homology axioms are essentially about pushouts). But the idea is that when the spaces are highly connected, π_n behaves like a homology theory. This follows from Blakers-Massey.

Look at Robert Graham's work on synthetic homotopy theory in HoTT; we can construct "regular homology" as an approximation to stable homotopy, in which suspension (i.e. smashing X with S^i), is replaced with smashing X with a truncation of S^i. $H_n^{reg}(X) = \lim_i \pi_{n+i}(X \wedge ||S^i||_i)$. Regular homology has a very similar definition with stable homotopy.

Definition 3.1. A *spectrum* is a sequence of simplicial sets

$$E = \{E_0, E_1, E_2, ...\}$$

together with structure maps $S^1 \wedge E_k \to E_{k+1}$ for $k \geq 0$. A *map of spectra* $f : E \to F$ is a sequence of maps $f_k : E_k \to F_k$ compatible with the structure maps in the sense of diagrams

$$\begin{tikzcd}
S^1 \wedge E_k \arrow[r] \arrow[d, "id_{S^1} \wedge f_k"'] & E_{k+1} \arrow[d, "f_{k+1}"] \\
S^1 \wedge F_k \arrow[r] & F_{k+1}
\end{tikzcd}$$

We let **Sp**(**Top**) denote the usual category of spectra of topological spaces.

Let $\Sigma^\infty X$ denote the usual suspension spectrum of X with $E_n = \Sigma^n X$, and let $\mathbb{S}$ denote the sphere spectrum with $E_0 = S^0$ and $E_n = \Sigma^n S^0$. Then, substituting $X = S^n$ into our definition of $[X, Y]_s$, we can define the (stable) homotopy groups of a spectrum:

Definition 3.2. The *(stable) homotopy groups* of a spectrum E are

$$\pi_q(E) = [\mathbb{S}, E]_q = \underset{k}{\text{colim}} \ \pi_{q+k}(E_k)$$

where the colimit is over the maps $\pi_{q+k}(E_k) \to \pi_{q+k}(\Omega E_{k+1}) \cong \pi_{q+k+1}(E_{k+1})$ for $k > -q$.

Another standard way of introducing and motivating spectra (for example, in [69, ?, ?]) is to describe the properties we want the stable homotopy category **Ho(Sp(Top))** to possess, e.g. it should have a suspension spectrum functor, be additive, be triangulated, satisfy Brown representability, etc.. Surprisingly (for someone interested in spectra because they help abstract cohomology) this is actually the most historically accurate description, at least in the years before Boardman finally gave a satisfactory construction of the category by applying colimits to finite CW spectra—the lead-up to a successful definition of the stable homotopy category lay precisely in the development and maturation of a language for expressing the requirements above (as well as a "hare"-like leap to embrace the new, abstract language). In its favor, the approach is also very convenient for proofs, which likely played and plays a role in its usage. It also respects the idea that there are many possible "categories of spectra"—of CW spectra, symmetric spectra, simplicial "pre"spectra (our definition above), S-modules—that descend to the same stable homotopy category **Ho(Sp(Top))**.

Question 14. How is the construction of new cohomology theories, using properties of **Ho(Sp(Top))**, connected to the construction of new spectral sequences which take groups in one cohomology theory then converges to another? I'm sure that it is, but the later constructions are still mysterious to me.

3.3 The derived setting

[I promise to fill this in once I finish my Homological Algebra class.] From the E-M axioms...

The theory of spectra is a language for stable homotopy, but with respect to actual computations

1. relate spectra to derived categories
2. basic motivation of hom and ⊗.
3. the derived functor construction
4. the derived category construction (ref. Thomas' note)

[?] [?]

[Below is Wikipedia's introduction of the derived setting, for reference.]

Definition 3.3. Let $A^\bullet$ be a complex in an abelian category. Define the kth *cohomology functor* by

$$H^k(A^\bullet) = \ker(d^k)/\mathrm{im}(d^{k+1})$$

where $d^k : A^{(k)} \to A^{(k-1)}$ is the coboundary map.

By virtue of being a functor, H^k satisifies the following diagram for any morphism u:

$$\begin{tikzcd}
A^\bullet \arrow[r, "H^k"] \arrow[d, "u"'] & H^k(A^\bullet) \arrow[d, "H^k(u)"] \\
B^\bullet \arrow[r, "H^k"'] & H^k(B^\bullet)
\end{tikzcd}$$

The properties of abelian categories guarantee that H^k is a cohomology theory, i.e. that it has a long exact sequence of homology and satisfies excision.

Definition 3.4. Let $\Gamma(X,-) : \mathcal{F} \to F(X), u \mapsto u_X$ be the global sections functor over X from sheaves of modules on X to modules). The *n-th sheaf cohomology functor* is

$$H^n_{\mathrm{Sh}}(X,F) := R^n\Gamma(X,F),$$

where $R^n\Gamma$ is the n-th right derived functor of Γ.

Grothendieck's definition of sheaf cohomology, now standard, uses the language of homological algebra. The essential point is to fix a topological space X and think of cohomology as a functor from sheaves of abelian groups on X to abelian groups. In more detail, start with the functor $E \to E(X)$ from sheaves of abelian groups on X to abelian groups. This is left exact, but in general not right exact. Then the groups $H^i(X,E)$ for integers j are defined as the right derived functors of the functor $E \to E(X)$. This makes it automatic that $H^i(X,E)$ is zero for $i < 0$, and that $H^0(X,E)$ is the group $E(X)$ of global sections. The long exact sequence above is also straightforward from this definition.

The definition of derived functors uses that the category of sheaves of abelian groups on any topological space X has enough injectives; that is, for every sheaf E there is an injective sheaf I with an injection $E \to I$. It follows that every sheaf E has an injective resolution:

$$0 \to E \to I_0 \to I_1 \to I_2 \to \cdots .$$

Then the sheaf cohomology groups Hi(X,E) are the cohomology groups (the kernel of one homomorphism modulo the image of the previous one) of the complex of abelian groups:

$$0 \to I_0(X) \to I_1(X) \to I_2(X) \to \cdots .$$

Standard arguments in homological algebra imply that these cohomology groups are independent of the choice of injective resolution of E.

The definition is rarely used directly to compute sheaf cohomology. It is nonetheless powerful, because it works in great generality (any sheaf on any topological space), and it easily implies the formal properties of sheaf cohomology, such as the long exact sequence above. For specific classes of spaces or sheaves, there are many tools for computing sheaf cohomology, some discussed below.

The chief problem in these cases is to construct a resolution (the "derived" or "acyclic" object) which is *functorial*.

Need a discussion of Whitehead's theorem on complexes. E.g. the "complexes good, homology bad" motto.

The chief problem...

However, the most geometrically-motivated approach comes from the homotopy category, and that story begins with the derived approach of Tor and Ext by Eilenberg; in the modern parlance, these are functors derived from the tensor product functor and the hom-functor, respectively. This language, while based in algebraic necessities, tends to obscure the geometric nature of the problem since all derived functor approaches come down to simplifying the analysis of certain resolutions, which are themselves ways of organizing the relations (and relations between relations, and so on) within a given set of geometric phenomena (cycles and boundaries, polynomial equations and syzygies, etc.). The derived functor is the functor defined on this n-categorical structure by the action of the original functor on the space. Since one can easily obtain the precise definition from the nLab or from Hatcher, I will not mention it here.

Question 15. In what way are spectral sequences "coordinatizations" of derived functors?

3.4 Model categories

Model categories were originally developed by Quillen in order to deal with the set-theoretic size issues arising from localization, for example in the homotopy category $\mathbf{Ho(C)}$, where we localize with respect to weak equivalences.[15]

Model categories are useful when we try to define notions of homotopy on categories beyond **Top**. In another direction, we can use ∞-groupoids (or ∞-categories) to do the same thing; in either case, the idea comes down to carrying the structure of **sSet**, the category of simplicial sets, to the category in which we want to do homotopy. We construct a synthetic setting where concepts like "path" and "point" are primitives, without inherent topological meaning.[16]

Definition 3.5. A (Quillen) *model category* is a category $\mathcal{M}$ together with three classes of morphisms, each closed under composition: *weak equivalences* W, *fibrations* Fib, and *cofibrations* Cof. The morphisms must satisfy the following axioms:

1. $\mathcal{M}$ is complete and cocomplete.

2. Two-out-of-three: if two out of the set f, g, and gf are in W, then so is the third.

[15] I'm told that a completely different approach was adopted by Grothendieck using Grothendieck universes.

[16] Perhaps a better word is "abstract" rather than synthetic. The categorical approach to paths and points is starkly different from that of synthetic geometry.

3. Retract: if g is a retract of h and h is in any of W, Fib, or Cof, then g is as well

4. Factorization: if $g : X \to Y$ is a morphism in $\mathcal{M}$, then it can be factorized as $f_g \circ i_g$ or $p_g \circ j_g$, where i_g is a "trivial" cofibration (meaning it is also a weak equivalence) and f_g is a fibration, and j_g is a cofibration and p_g is a "trivial fibration" (same as above).

The set $(W, \mathrm{Fib}, \mathrm{Cof})$ is called a *model structure* on $\mathcal{M}$.

The idea of the model category is that we are carrying additional data (in the form of Fib and Cof via the weak factorization) associated to the weak equivalences that allows us keep track of the higher homotopies associated to the weak equivalence; in this way, the object in the model category "models" the spaces (∞-groupoids) sitting in some ∞-category.[17]

Example 3.6. **sSet**, $\mathbf{sSet_\bullet}$, **Top**, $\mathbf{Top_\bullet}$, and **Sp(Top)** (to be discussed) are model categories.

As promised, we now give a model structure on the category of spectra.

Definition 3.7. A *stable equivalence* is a map of spectra $f : E \to F$ which induces an isomorphism on stable homotopy groups $\pi_i^s(E)$ and $\pi_i^s(F)$.

Note that one can invert all the stable equivalences, in which case we obtain the homotopy category of spectra **Ho(Sp(Top))**, e.g. the *stable homotopy category*, where all weak equivalences are now isomorphisms.

Definition 3.8. A *pointwise weak equivalence* (resp. fibration) of spectra is a map $E \to F$ such that for every $n \geq 0$ the map $E^n \to Y^n$ is a weak equivalence (resp. fibration) in **sSet**. A cofibration is a map with the left lifting property with respect to the maps that are both pointwise equivalences and pointwise fibrations.

This pointwise structure defines a model category structure on the category of spectra.

Definition 3.9. A *stable fibration* is a map in **Sp(Top)** with the right lifting property with respect to all cofibrations that are stable equivalences. The *stable structure* on **Sp(Top)** consists of the stable equivalences, the stable fibrations, and the cofibrations.

By proposition ? in Bousfield [**?**], the stable structure defines a model category on **Sp(Top)**.

[17] This setup serves the same purpose as the stack of identity types sitting above a type in Martin-Löf type theory (in fact, I believe Awodey and Warren constructed homotopy type theory this way, by defining Martin-Löf type theory in a model category). For more on the type-theoretic formalism, refer to Section A.

3.5 Brown representability

[This section is incomplete.]

Brown representability (and its earlier corollary, the characterization of Eilenberg-MacLane spaces as classifying spaces $\tilde{H}^n(X;G) \simeq [X, K(G,n)]$) are natural consequences of the cohomology axioms—one may regard them as immediate applications of the axioms.

We will first review Hatcher's proof for Ω-spectra, before considering Lurie's statement of Brown representability and (portions of) his proof.

Recall that an Ω-spectrum is a spectrum where the structure maps $E_k \to \Omega E_{k+1}$—here reformulated in terms of the loopspace functor Ω—are weak equivalences for all k. (Examples include the familiar Eilenberg-MacLane spectrum $HG = \{K(G,n)\}$ for various abelian groups G.) The following is Theorem 4.58 in Hatcher:

Theorem 3.10. *If $\{K_n\}$ is an Ω-spectrum, then the functors $X \mapsto h^n(X) = \langle X, K_n \rangle$, $n \in \mathbb{Z}$, define a reduced cohomology theory on the category of basepointed CW complexes with basepoint-preserving maps.*

The proof of this result comes down to checking the axioms and constructing the long exact sequence, and we do not reproduce it here. Brown representability states the converse: that every cohomology theory (in topology) arises from a spectrum. The following is Theorem 4E.1 in Hatcher:

Theorem 3.11 (Brown representability)**.** *Every reduced cohomology theory on the category of basepointed CW complexes and basepoint-preserving maps has the form $h^n(X) = \langle X, K_n \rangle$ for some Ω-spectrum $\{K_n\}$.*

Proof. Consider a single functor $h(X)$ satisfying the cohomology axioms; we want to show that it can be represented as $\langle X, K \rangle$ for some K. So we first show that the map

$$\begin{aligned} T_u : \langle X, K \rangle &\to h(X) \\ f &\mapsto f^*(u) \end{aligned}$$

is a bijection for a certain $u \in h(K)$. Hatcher shows this through a series of intervening lemmas; we will compress those in order to emphasize the main action of the proof, which is to illustrate the existence of a universal cohomology class $u \in h(K;G)$ whose pullback to $\langle X, K \rangle$ determines the bijection.

The proof comes down to verifying a series of commutative diagrams. Consider a cohomological functor h to abelian groups. Then there should exist a CW complex K for any choice of X such that the following diagrams hold.

$$\begin{tikzcd} (1) \quad h(K) \arrow[r, "f^*"] & h(X) \arrow[dl, leftrightarrow, "T_u"] \\ \langle X, K \rangle & \end{tikzcd}$$

First off we note that K must be connected.

Note that $T_u : \langle X, K\rangle \to h(X)$ is the map defined by $T_u(f) = f^*(u)$.

$$\begin{tikzcd}
(2) & (A,a) \arrow[r, "f"] \arrow[d, dashed, "i"'] & (K,u) \\
& (X,x) \arrow[ur, "g"'] &
\end{tikzcd}$$

It remains to stitch together all the K_n associated to a family of cohomology functors h^n, i.e. to demonstrate that each K_n is weakly equivalent to ΩK_{n+1} for all n.

□

[Discuss the strategy and reasoning behind Hatcher's approach.] As Ω-spectra are the cofibrant objects in the category of spectra, it generally suffices to show it for just the Ω-spectra.

...

By contrast,

In the ∞-category setting, Brown representability becomes

Now we consider Brown representability in the ∞-category setting. The following theorems and definitions are from Lurie [69].

A functor $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ is *representable* if there exists an object $X \in \mathcal{C}$ and a point $\eta \in F(X)$ which induces bijections $\hom_{\mathcal{C}}(Y, X) \to F(Y)$ for every object $Y \in \mathcal{C}$. Assuming $\mathcal{C}$ is *presentable*, that is ..., then the functor F is representable if and only if it takes colimits in $\mathcal{C}$ to limits in $\mathbf{Set}$.

Definition 3.12. An ∞-*category* (also known as a *weak Kan complex* [?] or a *quasi-category* [?]) is a simplicial set $\mathcal{C}$ which satisfies the following extension condition:

> Every map of simplicial sets $f_0 : \Lambda^n_i \to \mathcal{C}$ can be extended to an n-simplex $f : \Delta^n \to \mathcal{C}$, provided that $0 < i < n$.

Definition 3.13. A category is *presentable* if ...

Definition 3.14. Suppose $\mathcal{C}$ is a category with finite coproducts. An object $X \in \mathcal{C}$ is a *cogroup object* of $\mathcal{C}$ if it is equipped with a comultiplication $X \to X \cup X$ with the following property: for every object $Y \in \mathcal{C}$, the induced multiplication

$$\hom_{\mathcal{C}}(X,Y) \times \hom_{\mathcal{D}}(X,Y) \simeq \hom_{\mathcal{C}}(X \cup X, Y) \to \hom_{\mathcal{D}}(X,Y)$$

The following is Theorem 1.4.1.2. in Lurie [69]:

Theorem 3.15 (Brown representability)**.** *Let* $\boldsymbol{C}$ *be a presentable* ∞*-category containing a set of objects* $\{S_\alpha\}_{\alpha \in A}$ *with the following properties:*

1. *Each object* S_α *is a cogroup object of the homotopy category* ***Ho(C)****.*

2. *Each object $S_\alpha \in \boldsymbol{C}$ is compact.*

3. *The ∞-category $\boldsymbol{C}$ is generated by the objects S_α under small colimits.*

Then a functor $F : \boldsymbol{Ho(C)}^{op} \to \boldsymbol{Set}$ is representable if and only if it satisfies the following conditions:

(a) *for every collection of objects $C_\beta \in \boldsymbol{C}$, the map $F(\sqcup_\beta C_\beta) \to \prod_\beta F(C_\beta)$ is a bijection*

(b) *for every pushout square*

$$\begin{tikzcd} C \arrow[r] \arrow[d] & C' \arrow[d] \\ D \arrow[r] & D' \end{tikzcd}$$

in $\boldsymbol{C}$, the induced map $F(D') \to F(C') \times_{F(C)} F(D)$ is surjective.

Proof. That (a) and (b) follow from the representability of F is straightforward, since ... □

Remark 3.16. There's a certain kind of "is-just" explanation in category theory (or rather, in a certain line of pure category theory papers) which is sometimes jarring, like "Axiom J is just Yoneda". I can guess at why people do it, but there must be a better way of illustrating or presenting the categorical structure, that better motivates the presentation. Exposition matters!

3.6 A list of topological constructions

Let X be a topological space, i.e. a carrier set X endowed with a topology τ of open sets. All maps are continuous unless noted otherwise.

On spaces

The cone $CX = X \times I / X \times \{0\}$.

The suspension $SX = CX \cup_X CX$.

The reduced suspension $\Sigma(X)$ is the suspension of (X, x_0) with basepoint the equivalence class of $(x_0, 0)$.

The (Cartesian) product $X \times Y$.

The coproduct (aka wedge sum, aka 1-point union) $X \wedge X$. Key theorem is Van Kampen.

The smash product $X \vee Y = X \times Y / X \wedge Y$.

The join $X * Y$ is the space of all line segments joining points of X to Y, i.e. $X \times Y \times I / (x, y_1, 0) \sim (x, y_2, 0)$ and $(x_1, y, 1) \sim (x_2, y, 1)$. Worth noting that the join of n points is an $(n-1)$-simplex.

The pair (X, A) for A a subspace of X.

A covering space $\tilde{X}$ of X is (essentially) just the union of an open cover of X.

The loopspace ΩX is the space of (based) maps from S^1 to X. Key fact: $\pi_i(\Omega X) = \pi_{i+1}(X)$. More abstractly, $\Sigma(-) \dashv \Omega(-)$.

The free loopspace $\mathcal{L}X$ is the space of maps from S^1 to X.

The (James) reduced product (aka the free monoid generated by X) $J(X) =$.

The infinite symmetric product (aka the free abelian monoid generated by X) $SP(X) = J(X)/\sim_p = \bigcup_{n=1}^{\infty} SP_n(X)$ with the weak topology, where $SP_n(X) = X^n/\sim_p$ and $\sim_p$ is the equivalence relation we define for points that differ only by a permutation of coordinates. Note that $SP(X)$ is a commutative H-space, and the Dold-Thom theorem implies that SP is a functor taking Moore spaces $M(G,n)$ to Eilenberg-MacLane spaces $K(G,n)$.

On maps

Let $f : X \to Y$ be a continuous map.

The mapping cylinder $M_f = Y \cup_{(x,1)\sim f(x)} X \times I$.

The mapping cone $C_f = Y \cup_{(x,1)\sim f(x)} CX$.

The mapping torus for $f : X \to X$ is $T_f = X \times I/(x,0) \sim (f(x),1)$.

The suspension $Sf : SX \to SY$.

The homotopy fiber.

The Hopf invariant $H(f)$.

For complexes

The n-skeleton X^n is...

The Euler characteristic is the sum of even-dimensional simplices (or cells) minus odd-dimensional ones. Perhaps the most basic homotopy invariant.

Homology (where is Carlsson's formulation?).

Cohomology.

The Postnikov tower.

The (Quillen) plus construction.

For computing homology

There are many different ways to compute homology depending on the particular homology theory. Simplicial homology, for example, reduces down to rewriting a matrix quotient into Smith normal form [31] (assuming we have a basis for the space). Other questions like "what is $H_n(P^\infty)$?" or "what is $H_n(S^k \vee S^k)$?" or "compute $H^*(K(\mathbb{Z},n),\mathbb{Q})$ using a spectral sequence" or "find $H^n(V)$ of some projective variety" (all familiar from classroom assignments) involve more general techniques which depend on a more refined analysis of the pathspace and/or a particular decomposition of X (as a fibration, for example).

> "Here in fact we find a basic methodological characteristic common to all genuine statements of principles. Principles do not stand on the same level as laws, for the latter are statements concerning specific concrete phenomena. Principles are not themselves laws, but rules for seeking and finding laws. [...] The power and value of physical principles consists in this capacity for "synopsis," for a comprehensive view of whole domains of reality. Principles are invariably bold anticipations that justify themselves in what they accomplish by way of construction and inner organization of our total knowledge. They refer not directly to phenomena but to the form of the laws according to which we order these phenomena." - Ernst Cassirer, *Determinism and Indeterminism in Modern Physics*, 1956

4 Organizing principles in algebraic geometry

One thing that I am often told is that using category theory lets one avoid many redundancies, and that using it systematically gives one many efficiencies. "Build the theory and good things will happen," I am told. To which I counter, but doing anything systematically gives one many efficiencies. Pick a field that uses category theory extensively: algebraic geometry. What prompted a systematic approach in the first place? Or, if that question is too broad:

Question 16. What are the relevant aspects of classical algebraic geometry that made it useful to turn to a categorical language? And how do I translate, perhaps partially, these aspects to other fields?

A failed answer, to me, would be that the definitions of algebraic geometry were merely definable, in the sense that they could be written down in the language of whatever logic, even in the language of category theory.[18] Just because a theory is definable in a language does not mean it is *useful* to do so, else we would all be swamped with quantum-mechanical descriptions of Marxism. Just so, if I do not expect *set theory* to solve all the problems in my field, then why should I expect *category theory* to do the same?

A better but still insufficient answer would be that category theory is more efficient or "easier to swallow", in the sense of the old Atiyah quote, than whatever was there before in classical algebraic geometry.

> The aim of theory really is, to a great extent, that of systematically organizing past experience in such a way that the next generation, our students and their students and so on, will be able to absorb the essential aspects in as painless a way as possible, and this is the only way in which you can go on cumulatively building up any kind of scientific activity without eventually coming to a dead end. [5]

[18] Developed fully, I suppose this line of reasoning would lead to model theory, both as a manual for how to study a theory via definable equations, "the model theory of algebraic geometry", as well as its own exemplar of a field of study built on definable formulae, "model theory = algebraic geometry minus fields".

I don't doubt that the structural features of category theory play a role in managing the complexity of algebraic geometry, and that this role was valuable, even necessary for the development of modern algebraic geometry. But efficiency is not a sufficient explanation for Grothendieck's categorical reformulation of algebraic geometry nor was that reformulation based on entirely or even mostly on categorical ideas. "Applied category theory", in this case, wasn't just diagrams.

A successful answer would give us a selection of new results in algebraic geometry that came out of the categorical reformulation. It would show us the specific categorical mechanisms that allowed mathematicians to envision these new results and new proofs. It would tell us how the structure of classical algebraic geometry—the relationship between its open problems and its extant tools—invited and shaped the use of category theory. Most importantly, it would show us what was missing from category theory, and how those gaps were addressed.

The best answer would help us predict the success of applying category theory to a field and teach us how to adapt categorical and algebraic tools to the needs of different disciplines. It would tell us why category theory became fundamental for fields like algebraic geometry and algebraic topology but not (yet) for fields like combinatorics or probability theory.

I do not know why category theory proved so useful in algebraic geometry—if I did, I would not be asking the question! But I have a few allied intuitions, all centered around this idea: category theory is a *computational** lens on mathematics.[19]

(*) I have an unconventional definition of 'computational', so I have labeled it with an asterisk. A computational* theory is a theory that efficiently *abstracts out* questions of computability, complexity, and efficiency to focus on data access and manipulation. The fact that a computational* theory *does* abstract over the model of (normal) computation or deduction is important; see claim (1) below.

This intuition is based on progress connecting the theory of computation with computational applications in the sciences, e.g. see workshops at Berkeley (2002) and the IAS (2014); a network of connections between category theory and theoretical computer science, often through denotational semantics of models of computation; and some preliminary speculation relating categorical constructions to presentations of structured data in machine learning. (I believe that it was Emily Riehl who called category theory "a way of displaying the data, of all sorts".)

In what way was category theory a computational* lens on algebraic geometry? My best guess is that category theory

(1) clearly distinguished the computational*, accounting aspects of classical algebraic geometry from the underlying topological axiomatization (the Zariski topology),

[19] But not necessarily on the sciences, at least not without a mathematical account of what it means to have an "experiment".

(2) gave a coherent way of representing and abstracting over the accounting aspects via a new tool: cohomology theories with coefficients in a sheaf, and

(3) this sheaf cohomology filled a gap in category theory in that it allowed us to manipulate the computational* abstraction *in tandem* with the topological axiomatization.

The last assertion, (3), is the critical one. I modeled the first parts of my guess on May's [71] characterization of the Eilenberg-Moore axioms for cohomology, which "clearly and unambiguously separated the algebra from the topology", but I suspect that what makes cohomology in algebraic geometry "computational" (and not just in the sense of "do calculations", which happens anyway in algebraic topology) is the way in which it interacts with the topological axiomatization.

The assertion (3) may very well be wrong. It may not be at all productive to think of cohomology computationally*, of topology as separate from those aspects, or of sheaf theory and category theory as complementary. But I have to start somewhere. Based on my guess, I have reformulated Question 16 below.

Question 17. Knowing what a sheaf is and what it is for—namely, to set up the machinery for cohomology—what is the corresponding notion in classical algebraic geometry, i.e. in the language of rings, function fields, and rational maps?

I have split my answer to Question 17 into three parts: a review of sheaves, a review of sheaf cohomology (via Cech cohomology), and a preliminary discussion of the Weil conjectures and Weil cohomology. In Section 4.2, I form an explanation that begins with the classical versions of Riemann-Roch and genus over $\mathbb{C}$, and then show how sheaf cohomology facilitates extensions of Riemann-Roch to arbitrary fields (still algebraically-closed). In Sections 4.3-4.5, I review the cohomological machinery needed to prove this result, up to Serre duality. In Section 4.6, I then state that a good cohomology theory (in algebraic geometry) is something that imitates cohomology theories in algebraic topology, and that this means, for the purposes of number theory, that it ought to be a cohomological witness to the Weil conjectures. Finally, in Section 4.7, I explain what it means for a cohomology theory to witness the Weil conjectures, and how the Weil conjectures relate back to my characterization of category theory as a computational* lens on algebraic geometry.

Remark 4.1. Originally this note was meant to be a discussion of motives in algebraic geometry, following Voevodsky and Morel's own exposition in [117] and [83]. Since that was far too hopeful, I have tabled that discussion to Section G. Instead, I will give the very first part of that discussion on sheaves and topoi, following mainly Serre [96], with reference to historical notes of Gray [46] and Dieudonne [27] in addition to the class notes of Ritter and Vakil [114], an introductory note on sheaf theory by Lovering [67], an introductory note on the Weil conjectures by Osserman [87], the first three chapters of Hartshorne [49], and a large number of online sources, especially the nLab and the Stacks Project. Where possible, I have updated the definitions and notation of historical results to be consistent with the modern, categorical standard.

4.1 A very brief review of sheaf theory

The original *faisceau* (Leray, 1945 [62]) were built to capture variation in the fibers of a projection, as in a sheaf of modules over a topological space. The first example of a sheaf was the sheaf assigning to $U \subset X$ its pth cohomology group, i.e. an interpretation of Steenrod's cohomology with local coefficients and its use in the study of covering spaces of nonorientable manifolds (see Section 3.H of [50]). So the origins of sheaf theory were fundamentally topological. Leray later described operations on sheaves (images, quotients, limits, colimits, etc.), which were clarified and applied by Cartan and his students to great effect in complex analytic geometry. It was Serre, in the seminal "Faisceaux algébriques cohérents" (FAC) [96], who brought these methods from the Cartan seminaire to algebraic geometry; after Serre, the development of modern algebraic geometry is largely due to Grothendieck and his collaborators in SGA and EGA. For the most part, this note will deal with the 'liminal' algebraic geometry (1955-1960) between the publications of FAC and SGA, i.e. with the slightly easier exposition of sheaf cohomology on varieties rather than on schemes.

The most enlightening motto I have found for sheaf theory is this: sheaf theory defines the function theory of a space so that it respects the "local" structure at points. A manifold, for example, looks like Euclidean space near its points, so we would like functions defined on a manifold to behave, near points, like functions on Euclidean space. An algebraic variety looks like the zero set of some polynomials at its points, and we would like "regular functions" defined on the variety to behave like polynomials near its points. Much of sheaf theory comes down to making this idea precise.

Of course, not every "space" is a topological space, nor does every topological space have points (cf. locales), nor does every pointed topological space separate points (cf. the Zariski topology). To pre-empt these concerns, we will define sheaves on *sites* rather than on topological spaces. A *site* is a category C equipped with a "topology" τ defined in terms of sieves: collections of morphisms in C with codomain U that model the idea of a covering family over U. Where defined, τ is called the *Grothendieck topology* of C; I leave the exact definition of sieve and Grothendieck topology to the nLab.[20]

[20] I hope that I am not simply being obtuse in choosing to defines sheaves in term of sites. I have a tendency to view anything with a topology as an algebraic topologist would, but the most relevant aspect of the Zariski topology is that it is not algebraic topology in the same sense that a space like S^n is with its usual topology, cf. Example **??** or "topology is about semidecidable properties". The Zariski topology is a topological translation of something fundamentally algebraic, much like later "topological" axiomatizations such as the étale topology or the Nisnevich topology, which do in fact require the site-theoretic formulation. Just so, sheaves are not algebraic topology in the same sense that fibrations and spectral sequences are, no matter their common origins with Leray. Sheaves are specialized for spaces where the compositional and combinatorial assumptions of algebraic topology break down. [I keep going

In short, a presheaf is a contravariant functor to **Set**, and a sheaf is a presheaf $F : C^{\text{op}} \to \mathbf{Set}$ with an associated topology on C, such that the behavior of F on an "open set" U in C can be computed by gluing together the behavior of F on subsets of U. "Behavior", in the case of a sheaf, comes down to the existence and uniqueness of *sections*, i.e. elements of the set $F(U)$. Formally:

Definition 4.2. A *sheaf* is a presheaf F on a site (C, τ) satisfying the following axioms:

1. (Existence of gluing) Suppose the following conditions are satisfied: (1) $\{p_i : U_i \to U\}_{i \in I}$ is a covering family of $U \in \mathsf{Ob}\, C$ and (2) for each $i \in I$ we have a section $s_i \in F(U_i)$ such that on all "overlaps" $U_i \xleftarrow{f} K \xrightarrow{g} U_j$,
$$F(f)(s_i) =_{F(K)} F(g)(s_j).$$
Then there exists a section $s \in F(U)$ such that $F(p_i)(s) = s_i$ for every $i \in I$.

2. (Uniqueness of gluing) If $\{p_i : U_i \to U\}_{i \in I}$ is a covering family of $U \in \mathsf{Ob}\, C$, and if $s, t \in F(U)$ are sections such that $F(p_i)(s) =_{F(U_i)} F(p_i)(t)$ for all $i \in I$, then $s = t$.

In the context of a sheaf, the morphism $F(f) : F(V) \to F(U)$ induced on sections by a morphism $f : U \to V$ is called a *restriction map*, based on the intuition that we are restricting a function on V to a subset $U \subset V$. We sometimes call sections *local sections* to emphasize that they are the local components in a gluing. Any section constructed by gluing is called a *global section*.

Example 4.3. The skyscraper sheaf over U, which sends U to a set and everything else to the empty set.

Example 4.4. The sheaf of smooth functions on a real manifold. The fact that smooth functions form a sheaf should not be that surprising, since the sheaf structure merely quantifies the fact that smoothness is already a local-global property, namely smooth $\simeq$ locally smooth.

Example 4.5. The sheaf of analytic functions on a complex manifold.

Example 4.6. The sheaf of sections of a vector bundle $\pi : E \to B$, where the sections $S(U)$ are maps $s : U \to \pi^{-1}(U)$.

Example 4.7. The sheaf of germs of functions (on some object X) with values in an abelian group G, where the sections are functions $s : U \to G$ and the restriction maps are just function restriction.

Example 4.8. The sheaf of events of a quantum measurement scenario [1], or the sheaf of a choice scenario [122].

Example 4.9. The sheaf of sets on a poset with the Alexandroff topology, which is used to represent the functional dependencies of (dynamical) systems in [92].

back and forth on whether this statement is right or wrong.]

Example 4.10. Continuous, discrete, and synchronous interval sheaves corresponding to different notions of time in dynamical systems, which are conceived as spans over such sheaves [104].

Example 4.11. The constant presheaf, which sends every U identically to a set A, is *not* typically a sheaf. By the first sheaf condition, any two local sections $a_i, a_j \in A$ defined on disconnected open sets U_i, U_j, respectively, should glue into a global section since they agree trivially on "all overlaps" $U_i \cap U_j = \emptyset$. However, this global section cannot exist in $F(U_i \cup U_j) = A$ if $a_i \neq_A a_j$.

There is a special language for sheaves when $C = \mathbf{Op}(X)$, the category of open sets and inclusions of a topological space X.

Definition 4.12. Let F be a sheaf (or presheaf) on $\mathbf{Op}(X)$ for some topological space X, and suppose x is a point in X. Let $U, V \in \mathbf{Op}(X)$ and $S = \bigsqcup_{U \ni x} F(U)$ be their disjoint union. Then:

1. The *stalk F_x of F at x* is the set of equivalence classes in S under the relation $f \sim g$, where $f \sim g$ if there exists some open $W \subset U \cap V$ containing x such that $f|_W = g|_W$. Categorically, F_x is the colimit over the system of objects $\{F(U) : x \in U\}$ where the arrows are the restriction maps between objects.

2. The equivalence classes of a stalk are called the *germs* of the stalk. Intuitively, germs are defined in a small neighborhood of x, and can be used to compute local properties like the derivative of a section but none of the actual values of the section anywhere else besides x.

3. A *section* $s \in F(U)$ can be interpreted as a map $s : U \to \mathcal{F}$. There are natural projection maps $s \mapsto s_x$ taking sections to germs.

4. For an inclusion $i : V \hookrightarrow U$, the restriction map on sections is just function restriction.

$$\begin{aligned} F(i) : F(U) &\to F(V) \\ s &\mapsto s|_V \end{aligned}$$

5. The *étale space* of a sheaf F (but not a presheaf) is $\mathcal{F} = \coprod_{x \in X} F_x$. $\mathcal{F}$ has a topology inherited from X taking the $F(U)$ as a basis, which motivates studying its topological features such as (co)homology.

Example 4.13. The *sheafification* of a presheaf is the left adjoint to the inclusion from $\mathbf{Sh} \to \mathbf{PSh}$. Intuitively, we can think of sheafification as a process of adding all the global sections that are missing from the presheaf, and of deleting all the non-zero global sections that are locally zero.

Sheafification is an extremely important part of the machinery of sheaf theory. It is quite literally the process by which one conforms some global property defined on sections, like being constant, onto a topological structure.

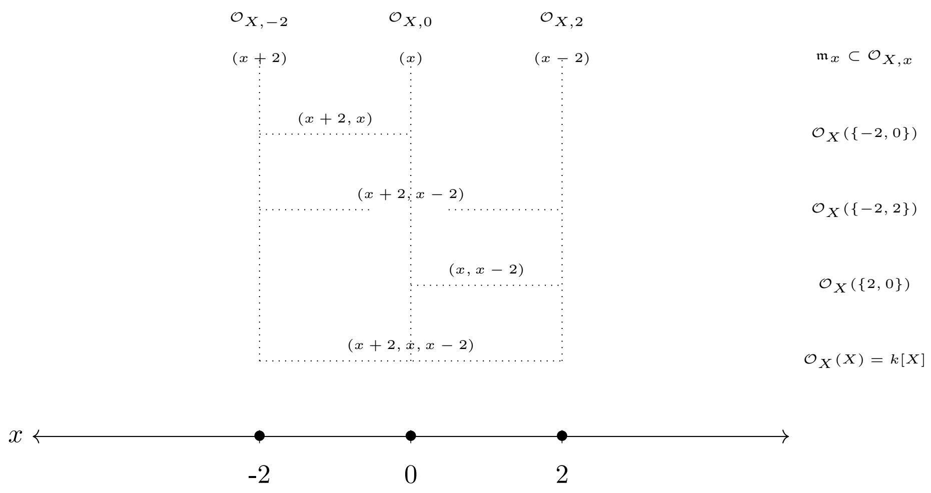

$\mathcal{O}_{X,-2}$
$\mathcal{O}_{X,0}$
$\mathcal{O}_{X,2}$
$(x+2)$
(x)
$(x-2)$
$\mathfrak{m}_x \subset \mathcal{O}_{X,x}$
$(x+2, x)$
$\mathcal{O}_X(\{-2, 0\})$
$(x+2, x-2)$
$\mathcal{O}_X(\{-2, 2\})$
$(x, x-2)$
$\mathcal{O}_X(\{2, 0\})$
$(x+2, x, x-2)$
$\mathcal{O}_X(X) = k[X]$
x
-2
0
2

Figure 4.1: A representation of the sheaf of regular functions $\mathcal{O}_X$ over $X = \mathbb{V}(x^3 - 4x)$, based on David Mumford's sketch of the sheaf over $\operatorname{Spec} \mathbb{Z}[x]$ in *The Red Book of Varieties and Schemes*. Vertical lines represent stalks over points. Horizontal lines represent (elements of the étale space corresponding to) global sections over subvarieties of X.

Example 4.14. The constant sheaf on a topological space X with value A, A_X, is defined as the sheafification of the constant presheaf on X with value A. It can be obtained by adding all the global sections defined by gluing elements of $a \in A$ across disconnected open sets. Assuming that the underlying site has colimits (and thus stalks), the constant sheaf is identically A on stalks and has "locally constant" global sections, i.e. constant over connected components.

Example 4.15. The sheaf of regular functions $\mathcal{O}_X$ on an affine variety X, a.k.a. the sheafification of the presheaf of (globally) rational functions on X.

Example 4.16. More generally, $\mathcal{O}_X$ is an example of a sheaf called the *structure sheaf* $\mathcal{O}$ of a locally ringed space $(X, \mathcal{O})$.

Example 4.17. The sheaf of rational functions $\mathcal{K}_X$ on an affine variety X, not to be confused with the sheaf of "locally rational functions" $\mathcal{O}_X$ above, which is a subsheaf of $\mathcal{K}_X$. On affine varieties (i.e. integral schemes), $\mathcal{K}_X$ is a constant sheaf, and the stalk $\mathcal{K}_{X,x}$ over any point in X corresponds to the function field of X. On general schemes, $\mathcal{K}_X$ leads to the field of birational geometry.

My prototypical example of a sheaf, following Serre, is the structure sheaf of an affine variety X over a field k, $\mathcal{O}_X : \mathbf{Op}(X) \to \mathbf{Ring}$. $\mathcal{O}_X$ takes Zariski-open sets of X to rings and inclusions to ring homomorphisms, such that the stalk of $\mathcal{O}_X$ over x

$$\mathcal{O}_{X,x} = \operatorname*{colim}_{U \ni x} F(U)$$

is a local ring, called the *localization of* $k[X]$ *at* x. Serre calls $\mathcal{O}_X$ the *sheaf of* *local rings* of X; in Hartshorne, $\mathcal{O}_X$ is known as the *sheaf of regular functions on* X. This definition is surprisingly subtle, so we will go through the idea a few times. Also, since we are defining the sheaf by its sections (instead of through some other process, cf. Example 4.15), we will need to check that it actually satisfies the sheaf axioms.

In the language of regular functions: the localization of $k[X]$ at x, i.e. the stalk $\mathcal{O}_{X,x}$, is the ring of germs of regular functions at x, where a function is *regular at* x if, in some open neighborhood $W \subset U$ containing x, it is equal to a quotient of polynomials p, q, with $p, q \in k[X]$ and $q(w) \neq 0$ for all $w \in W$. The particular quotient p/q is allowed to range over different open sets; i.e. a fixed regular function or "local section" may be defined by multiple rational functions over different open sets; they must only agree on their overlap. The ring of regular functions $\mathcal{O}_X(U)$ on U is the set of functions which are regular at all $x \in U$; we say that such functions are *regular on* U.

Geometrically: by construction, a regular function on X is something like a piecewise-rational function where the "pieces" are distinguished affine open sets in X. Localization at x collapses all the regular functions which are equal on a small-enough affine piece around x into a single germ at x—I have the image of compressing many parallel fibers into a single thread (without any twisting, for now). The set of such germs is a local ring because there is an obvious unique maximal ideal that "sucks in" all the rest: the ideal of all polynomials that are 0 at x.

Intuitively, the local rings are the abstract carriers of the "geometry", so that as we zoom in from any open set $U \ni x$ to the point x by following the directed colimit, the ring structure of $\mathcal{O}_{X,x}$ will recover the original geometric picture of the zero-set of a bunch of polynomial equations. This is analogous to the way that, as we zoom in to any point on a manifold, the smooth structure of the manifold will recover the structure of Euclidean space. One can think of the sheaf structure as a set of comprehensive instructions on how to zoom in—just follow the restriction maps!

Algebraically: in short, recall that

- the vanishing ideal $\mathbb{I}(X)$ = all polynomials that are identically 0 on X,
- the coordinate ring $k[X] \simeq k[x_1, ..., x_n]/\mathbb{I}(X)$ = polynomial functions restricted to X and identified by their images,
- the localization of $k[X]$ at a multiplicative set $S \subset k[X]$ is a new ring

  $$k[X][S^{-1}] := k[X] \times S/\sim$$

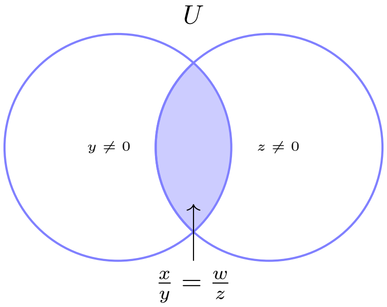
  where $(p, s) \sim (p', s')$ iff $ps' - p's = 0$.[21]
- for a given point $x \in X$, the localization of $k[X]$ at x is the localization of $k[X]$ at $Q = \{q \in k[X] : q(x) \neq 0\} = k[X] \setminus \mathfrak{m}_x$, where $\mathfrak{m}_x$ is the maximal ideal corresponding to x.

[21] $k[X]$ is an integral domain. More generally, we can localize any ring at S, for which we require that $(p, s) \sim (p', s')$ iff $t(ps' - p's) = 0$ for some $t \in S$.

- The localization of $k[X]$ at an open set $U \subset X$ is the localization at $Q = \{q \in k[X] : \forall x \in U, q(x) \neq 0\}$. This is *not* always equal to $\mathcal{O}_X(U)$, see the next example.

Where X is an irreducible affine variety, $k[X]$ is an integral domain, and the localization $S^{-1}k[X]$ (also written as $k[X][S^{-1}]$) is just the quotient field $k(X)$, called the *function field of* X.

Example 4.18. When $U = D_f$ is a basic open set in an affine variety X, then $\mathcal{O}_X(U)$ is the localization of $k[X]$ at D_f. But not every ring of regular functions $\mathcal{O}_X(U)$ arises as a localization. Suppose $X = \mathbb{V}(xz - yw)$, and let $U \subset X$ be the open subset of X where $y \neq 0$ or $z \neq 0$. Then the function x/y is defined on the subset of U where $y \neq 0$, the function w/z is defined on the subset of U where $z \neq 0$, and by the equation defining X, these functions are equal when they are both defined.



But there is no way to represent the two functions x/y and w/z as a single quotient of polynomials in $k[x, y, w, z]$ defined on all of U, even though they both belong to the same germ in $\mathcal{O}_{X,p}$ for $p \in W = \{p \in X : y \neq 0 \text{ and } z \neq 0\}$.

Categorically: the localization of $k[X]$ at a multiplicative subset S is a new ring $k[X][S^{-1}]$ in which all the elements of S become invertible, and which is universal with respect to this property. In detail: consider the category of $k[X]$-algebras (here, R-algebra means an R-module with an associate R-bilinear operation with a multiplicative identity). First off, $k[X]$ is itself a $k[X]$-algebra, and for any $k[X]$-algebra A, there is a canonical homomorphism $k[X] \to A$ defined by $p \mapsto p \cdot \mathbf{1}$. Second, for any multiplicative set $S \subset k[X]$, there is a subcategory of $k[X]$-algebras where we restrict to just those algebras A such that, under the canonical homomorphism $k[X] \to A$, every element of S is sent to an invertible element in A. Then the localization of a ring $k[X]$ at S is the initial object in this special subcategory of $k[X]$-algebras.

Localization is important because it is a tool for basing algebraic constructs within sheaf theory. I hope to come back to it in a later note on analogs of localization in machine learning.

Lemma 4.19. *For an affine variety X, $\mathcal{O}_X$ is a sheaf.*

Proof, version 1. Clearly $\mathcal{O}_X$ is a presheaf, so we will check the sheaf axioms directly.

Say that we have a set of regular functions $\{f_i\}_{i \in I}$ for all subsets U_i of a cover of U such that they agree on all overlaps. Let $f = (f_i)$ be the function defined by gluing them together in the obvious way. It is clearly regular on U, since the condition $f = p/q$ is only enforced locally; i.e. it can be satisfied by different p, q at different $p \in U$. This proves (1).

Suppose that $s, t : U \to k$ are regular functions that agree on all the open sets U_i in a cover of U. Then by virtue of being a cover of U, they agree on all U and thus belong to the same germ $\mathcal{O}_{X,x}$ for all $x \in U$. This proves (2). □

Proof, version 2. $\mathcal{O}_X$ is the sheafification of the presheaf of rational functions, denoted $\mathcal{O}'_X$ and defined by

$$\mathcal{O}'_X(U) = \{f : U \to k : \exists p, q \in k[X], q(x) \neq 0, \text{ s.t. } f(x) = \frac{p(x)}{q(x)} \; \forall x \in U\}.$$

$\mathcal{O}'_X$ is a presheaf and not a sheaf since not every ring of sections arises as the localization of $k[X]$ at U, as shown in Example 4.18. □

Proof, version 3. The following is Serre's original construction of $\mathcal{O}_X$.

Let k be a field, and let k^n be the affine space of dimension n, endowed with the Zariski topology. Pick $x = (x_1, ..., x_n) \in k^n$, and define the local ring of x, $\mathcal{O}_{k^n,x}$, in the usual way. These local rings define a subsheaf $\mathcal{O}_{k^n}$ of the sheaf F_{k^n} of germs of functions on k^n with values in k (see Example 4.7).

Define a closed set in k^n (i.e. an affine variety), and let F_X be the sheaf of germs of functions on X with values in k. For $x \in X$, this defines a canonical homomorphism on stalks,

$$\epsilon_x : F_{k^n,x} \to F_{X,x}.$$

The image of $\mathcal{O}_{k^n,x}$ under ϵ_x defines the stalks of a subsheaf of F_X which we denote $\mathcal{F}_X$. $\mathcal{F}_X$ is obviously a sheaf. It remains to prove that its stalks are isomorphic to the localization of the coordinate ring $k[X]$ at x, i.e. $\mathcal{O}_{X,x}$. But this follows directly from the fact that the kernel of ϵ_x is the ideal $\mathbb{I}(X) \cdot \mathcal{O}_{k^n,x}$, whose objects are just the regular functions that vanish on X. □

The structure sheaf $\mathcal{O}$, along with the closely related idea of ringed spaces, is the basic data structure used by all of modern algebraic geometry.

Definition 4.20. A *ringed space* (X, S) is a set X along with an associated sheaf of rings, S.

Example 4.21. An affine variety X with its structure sheaf $\mathcal{O}_X$ forms a ringed space, $(X, \mathcal{O}_X)$.

In Question 17, I have already claimed that sheaves are designed for cohomology. Similarly, we will see that a ringed space can be thought of as a space that comes with its own notion of cohomology.

Actual computations in sheaf theory require further manipulations of $\mathcal{O}$, namely subsheaves, direct image sheaves, quotient sheaves, and sheaves of $\mathcal{O}_X$-modules. In particular, sheaves of $\mathcal{O}_X$-modules will play an important role in defining the right sort of linear algebra over sheaves.

Definition 4.22. A *sheaf of $\mathcal{O}_X$-modules* is a sheaf F such that, for any $U \subset X$, $F(U)$ is an $\mathcal{O}_X$-module such that the restriction maps $F(U) \to F(V)$ are compatible with the usual restriction maps in $\mathcal{O}_X(U) \to \mathcal{O}_X(V)$ in the sense that $(fs)|_V = f|_V s|_V$ for all $f \in F(U)$ and $s \in \mathcal{O}_X$.

(I must admit that the multiplication here was a bit disorienting at first—if algebra over k is a little like recording chess moves with the numbers of k, then algebra over a sheaf is a bit like chess where pieces of the board are constantly being added or subtracted.)

Example 4.23. An *ideal sheaf* $\mathcal{I}$ of $\mathcal{O}_X$ is any sheaf of $\mathcal{O}_X$-submodules. In particular, $\mathcal{O}_X(U) \otimes \mathcal{I}(U) \subset \mathcal{I}(U)$ for any open U, and $I(U)$ is an ideal in $\mathcal{O}_X(U)$.

Example 4.24. For A a sheaf of rings and two sheaves of A-modules F, G, the *tensor product of F and G* is a sheaf $F \otimes_A G$ defined on stalks by $F_x \otimes_{A_x} G_x$. If the stalks A_x are commutative, then the tensor product is also a sheaf of A-modules.

I will hold off on any further definitions for now. The point is that, in reference to Question 17, $\mathcal{O}_X$ reproduces the objects of classical algebraic geometry, e.g. coordinate rings ($\mathcal{O}_X(X)$), function fields (the sheaf of rational functions[22]), and rational maps (morphisms of sheaves). See Table 4.1 for a complete comparison. However, we have yet to reproduce "what sheaves are for"—namely cohomology—in the language of classical algebraic geometry.

4.2 Good cohomology, part 1

Serre's central motivation in FAC was the search for a good cohomology theory for abstract varieties. For example, he cites as inspiration work by Kodaira and Spencer generalizing the Riemann-Roch theorem over complex algebraic curves to abstract varieties in the sense of Weil. There's a lot to unpack in this statement, so let me go through it slowly.

Consider the following very general problem: how can we read the topology of a variety X in terms of the polynomials which define it? In dimension one, this answer is just "by the number of roots". In dimensions two (and higher), the answer is essentially "by its genus." This is the genius of Riemann-Roch.

In the original version by Riemann and his student Gustav Roch, Riemann-Roch relates the genus[23] of a compact Riemann surface S to the function theory over (formal sums of) points of the surface. This should sound familiar—remember that sheaf theory is all about constraining the function theory of a space so that it can be reconstructed from the functions around points. Riemann's basic idea "begins by essentially creating the topological study of compact oriented surfaces, attaching to each surface S an invariantly defined integer

[22] A piece of confusing notation; the name comes from the nLab. The sheaf of rational functions should not to be confused with the presheaf of "globally rational functions" whose sheafification is the sheaf of regular functions, a.k.a. the sheaf of "locally rational functions". Serre calls the sheaf of rational functions the *sheaf of fields*.

[23] I'll only consider complex, non-singular projective curves in this section, so I won't distinguish the topological, geometric, and arithmetic genus.

geometry	algebra	sheaf	scheme
affine variety $X \subset k^n$	coordinate ring $k[X]$	global sections $\mathcal{O}_X(X)$	affine scheme Spec A
polynomial function $X \to k$	k-algebra hom $k[Y] \to k$	constant sheaf with value k	
polynomial map $X \to Y$	k-algebra hom $k[Y] \to k[X]$	morphism of sheaves $\mathcal{O}(Y) \to \mathcal{O}(X)$	
† point $x \in k^n$	maximals ideals $\mathfrak{m}_x \in \mathrm{Specm}(k[x_1, ..., x_n])$	stalk $\mathcal{O}_{X,x}$	functor of points?
projective variety X	homogeneous coordinate ring $S(X)$		
function field $k(X)$	f.g. field extension of k	sheaf of rational functions K_X	local ring of generic point
* (dominant) rational map $X \to Y$	k-algebra hom $k(Y) \to k(X)$	morphism of algebraic varieties	morphism of schemes
open covering $\mathfrak{U}$	nerve or Cech complex $\mathcal{N}(\mathfrak{U})$	Cech cocomplex	étale space
gluing?	linear algebra	sheaf cohomology	étale cohomology
affine space k^n	f.d. vector space	coherent sheaves	coherent sheaves
genus of a curve	genus of a number field	$H^1_{\mathrm{Weil}}(X, F)$	
Euler-Poincaré characteristic	Riemann-Roch	Serre duality	coherent duality
geometric dimension	Krull dimension	Krull dimension	Krull dimension
[a] algebraic spaces **Ring** → **Set**	presheaves **Ring** → **Set**	sheaves **Ring** → **Set**	functor of points

Table 4.1: Glossary of algebraic geometry. All varieties are irreducible. * indicates an equivalence of categories. † indicates a bijection.

[a] There are really three categories in the functor of points perspective: the category of functors from **Ring** to **Set**, the subcategory of sheaves within this category, and finally the further subcategory of algebraic spaces within the category of sheaves. Based on a comment of James Borger [56], we can think of these categories as corresponding to set theory, topology, and geometry.

$2g$, the minimal number of simple closed curves C_j on S needed to make the complement S' of their union simply connected" [27, pg. 836]. (Of course $2g$ is just the first Betti number, i.e. the dimension of $H_1(S, \mathbb{C})$.) This led Riemann, after some analytic arguments, to consider the field of rational functions over S, as he observed that the genus g was invariant under *birational transformations* of curves.

It's hard to overstate how important Riemann's observation was to the development of algebraic geometry. Besides leading directly to the Riemann-Roch theorem and later to the Riemann-Hurwitz formula, Riemann also considered parameterizations of the algebraic curves of a fixed genus, which would lead to the modern theory of moduli spaces of curves. The use of genus as an birational invariant introduced an extremely fruitful model for importing other topological ideas into geometry, later culminating in the development of homological algebra.

Before stating the Riemann-Roch theorem, I should refresh an important concept from commutative algebra. We know what an ideal of a ring is. In particular, ideals of the coordinate ring correspond to sets of *finite* points of an affine variety. A *divisor* can be thought of as an ideal "considered along with the point at infinity", since they account for the additional data at singularities. Suppose K is a finite extension of the field $\mathbb{C}(x)$ of rational functions in one variable over $\mathbb{C}$. Riemann's observation tells us that K corresponds to a class of Riemann surfaces. Then a *divisor D on K* is a formal sum $a_1x_1 + ... + a_nx_n$ of a finite set of points of X with integer coefficients, and the set of divisors $\mathcal{D}(K)$ of K is an additive group isomorphic to $\mathbb{Z}^X$. The *degree of a divisor*, $\deg(D)$, is the sum

$$\sum_{i=1}^{n} a_i$$

and the *support* of D is the set of $x_i \in X$ such that $a_i \neq 0$. D is *positive* if $a_i \geq 0$ for all a_i, and *negative* if $a_i \leq 0$ for all a_i.

A divisor D on K can be taken to represent a finite set of zeroes and poles of order $a_1, ..., a_n$ at prescribed points $x_1, ..., x_n$ (on a compact Riemann surface X with function field K). In particular, a *principal divisor* on K is a divisor of the form

$$(f) = \sum_{x \in X} v_x(f)x$$

where f is a meromorphic function and

$$v_x(f) = \begin{cases} a & x \text{ is a zero of order } a \\ -a & x \text{ is a pole of order } a \end{cases}$$

The principal divisors on K form a subgroup of $\mathcal{D}(K)$. Two divisors that differ by the addition of a principal divisor are called *linearly equivalent*; in particular $\deg(D) = \deg(D')$ if D is linearly equivalent to D'. In addition, for any surface there is a *canonical divisor* $\Delta = (\partial)$ of a nonzero meromorphic 1-form ∂ on the surface; this is canonical since any two nonzero meromorphic 1-forms will yield linearly equivalent divisors.

In Riemann-Roch, the notion of divisor represents not a particular set of zeroes and poles but a class of constraints on the topology. The divisor and degree of a divisor form one sort of "computational" interface between the local data at points and the global invariant, the genus. Suppose $D = \sum a_i x_i$, not necessarily a principal divisor. Riemann found that if $\deg D = \sum a_i \geq g+1$, then there exist rational functions on X with poles of order a_i for *any* set of points $x_i \in X$. Alternately, if $\sum a_i \leq g$, then such rational functions exist only for a set of special points, i.e. only for particular divisors D with those coefficients. (To avoid confusion: here the positive coefficients a_i of D control the order of the poles while the negative ones control the zeroes; see below.) The result is an application of Riemann-Roch; to state it, we first use D to define a complex vector subspace $L(D) \subset K$ of rational functions $f \in K$ on X such that

$$(f) + D \geq 0.$$

Explicitly, this means that $f \in L(D)$ is a rational function with poles of order at most a_i (if a_i is positive) and zeroes of order at least $-a_i$ (if $-a_i$ is negative). Let $\ell(D)$ be the dimension of $L(D)$. Then:

Theorem 4.25 (Riemann-Roch for compact Riemann surfaces)**.** *For a compact Riemann surface of genus g with canonical divisor Δ,*

$$\ell(D) - \ell(\Delta - D) = \deg(D) - g + 1.$$

If D is a principal divisor (in particular, if it is the principal divisor for a rational function), then $\ell(\Delta - D)$ is 0. Then if $\deg D \geq g+1$, the theorem implies that $\ell(D) \geq 2$. Supposing D is non-zero, then $L(D)$ always contains the constant functions (which occupy the first dimension), and saying that $L(D) \geq 2$ means that it contains a non-constant rational function with the prescribed poles (and no zeroes). Since we showed this using only the degree of D, such a rational function exists for any set of points on the surface.

The classical Riemann-Roch is stated for compact Riemann surfaces, which are equivalent to non-singular projective curves over $\mathbb{C}$.[24] To generalize to curves over arbitrary fields (still algebraically-closed), we need to shift away from the language of meromorphic functions over $\mathbb{C}$ to the general theory of rational functions over k. Just to reiterate, there are two things going on here:

(1) a change of base field from $\mathbb{C}$ to some other algebraically-closed field, e.g. the field of algebraic numbers, and[25]

(2) a change in language from meromorphic functions to rational functions—in particular, from analytic theorems about differential forms and abelian integrals to homological theorems about cycles and the dimensions of certain cohomology theories.

[24] Essentially, we want non-singular because it allows us to use Poincaré duality. We want projective because these correspond to the "compact" cases of varieties.

[25] What can we model, in machine learning, as a change of basis?

In my initial hypothesis for Question 16, I claimed that category theory, extended by sheaf theory, allowed us to co-evolve the computational abstraction in tandem with the particular (topological) axiomatization. Concretely, changing the computational abstraction corresponds to (2) and is elegantly captured in the idea of a morphism of (often, exact sequence of) sheaves, while changing the topological axiomatization corresponds to (1) and can be formalized via the base change functor. Both historically and substantively, the Riemann-Roch theorem gives us evidence of a correspondence between (1) and (2).

Note that there is also a third, separate aspect:

(3) a transformation of spaces, e.g. from X to its (open) subsets, especially when modeling a gluing.

I have tried to work out the motivation for (1) in Section 4.7, but, to be frank, I still do not completely understand the reason for (2). I have not worked out the specifics of Riemann's proof of Riemann-Roch, so it's hard for me to see the precise analytic arguments cohomology is replacing. Indeed, almost all modern introductions to Riemann-Roch feature the homological version, differing only in the amount of analytic versus homological material (e.g. compare Serre's to Vakil's to Hartshorne's). For me, the more analytic versions seem more elementary, and give the essential flavor of sheaf cohomology. The following is from Vakil [114].

Theorem 4.26 (Riemann-Roch, homological version)**.** *For C a non-singular projective curve over an algebraically-closed field, k, and $\mathcal{L}$ an invertible sheaf of degree d on C, then*

$$h^0(C, \mathcal{L}) - h^0(C, \Omega^1_C \otimes \mathcal{L}^\vee) = d - h^1(C, \mathcal{O}_C) + 1,$$

where $\mathcal{L}^\vee$ is the monoidal dual to $\mathcal{L}$ and Ω^1_C is the cotangent sheaf of C. h^0, h^1 correspond to the dimensions of their respective cohomology groups.

I leave the statement of the theorem "on the board" here as motivation. The key to the theorem, as well as to a vast range of later generalizations of Riemann-Roch[26] to higher dimensions and to arbitrary fields, is the cohomology over sheaves, and in particular Serre duality, which is an analog of Poincaré duality ($H^i(X) \simeq H^{n-i}(X)$) for sheaf cohomology and perhaps the key technical result coming out of FAC.

[26] E.g. Hirzebruch-Riemann-Roch and Grothendieck-Riemann-Roch. Note how Toen (in lecture) notes the distinction between topological and algebraic invariants on the two sides of the Hodge decomposition in H-R-R: $H^i(X, \mathbb{Q}) \otimes \mathbb{C} \simeq \bigoplus_{p+q=i} H^i(X, \Omega^q_X)$, as well as on the two sides of the étale ℓ-adic cohomology, $H^i_{et}(X, \mathbb{Q}_\ell) \simeq$ sheaf cohomology for the Zariski topology. An additional example comes from Kodaira, who lifted the work in terms of divisors into the language of line bundles and the Hodge theory of harmonic forms (itself a modernization of the analytic arguments used by Riemann to characterize the genus as a birational invariant), obtaining a Riemann-Roch formula for compact Kähler manifolds. In particular, the Kodaira embedding theorem (1954) characterizes non-singular, complex projective curves as the class of compact Kähler manifolds endowed with a kind of cohomological invariant called a *Hodge metric*. If I want to learn more Hodge theory, this would be an interesting place to start.

To understand the homological translation of Riemann-Roch and to prove Serre duality, we will need two ingredients: sheaf cohomology for "linearizing" the function space of an algebraic variety, and coherent algebraic sheaves, the sheaf-theoretic analog of finite-dimensional vector spaces.

4.3 A very brief review of sheaf cohomology

In this section, I will develop the preliminaries of sheaf cohomology via Cech cohomology, an approximation of sheaf cohomology based on finite coverings, roughly analogous to the simplicial division we use in algebraic topology.

Let $\mathfrak{U} = \{U_i\}_{i \in I}$ be an open covering of X. If $(i_0, ..., i_p)$ is a finite sequence of elements of I, we put

$$U_{i_0 \dots i_p} = U_{i_0} \cap ... \cap U_{i_p}.$$

From algebraic topology, recall that the *nerve* (or Cech complex) of a covering $\mathfrak{U} = \{U_i\}_{i \in I}$ is the abstract simplicial complex $\mathcal{N}(\mathfrak{U})$ with vertex set I, where a family $\{i_0, ..., i_p\}$ spans a p-simplex σ if and only if $U_{i_0} \cap \cdots \cap U_{i_p} \neq \emptyset$. In that case, we say that $U_{i_0} \cap \cdots \cap U_{i_p}$ is the *support* of σ. Let $K_p(I)$ be the free group generated by the set of p-simplexes. The boundary map in the simplicial complex is the usual $\partial : K_{p+1}(I) \to K_p$ is defined by the formula

$$\partial(i_0, ..., i_{p+1} = \sum_{j=0}^{p+1} (-1)^j (i_0, ..., \hat{i}_j, ..., i_{p+1})$$

where, as always, $\hat{i}_j$ means that the term i_j should be removed from the sequence.

The following definitions are from Serre.

Definition 4.27. Let F be a sheaf of abelian groups on X. If p is an integer ≥ 0, we call a *Cech p-cochain of* $\mathfrak{U}$ to be a "function" (note the variable codomain)

$$\begin{aligned} f : K_p(I) &\to F(U_{i_0 \dots i_p}) \\ (i_0, ..., i_p) &\mapsto f_{i_0 \dots i_p} \end{aligned}$$

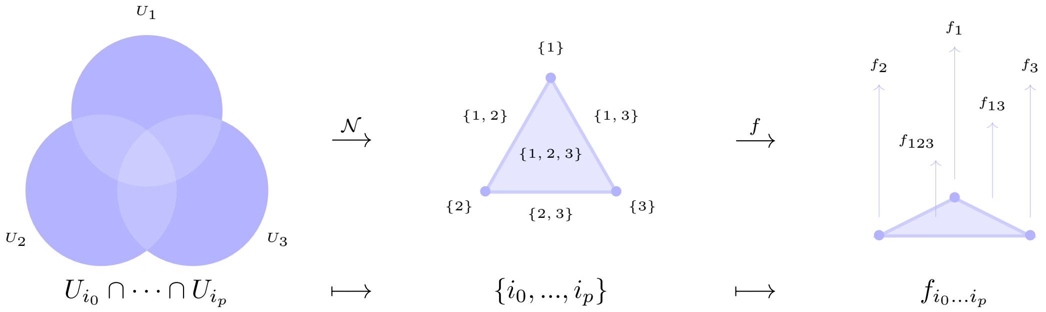
where $K_p(I)$ denotes the chains of dimension p.

Definition 4.28. Given any sheaf of abelian groups F, the *p-th Cech cochain group of* $\mathfrak{U}$ *over* F is the product

$$C^p_{\text{Cech}}(\mathfrak{U}, F) = \prod_{i_0 < \dots < i_p} F(U_{i_0} \cap \cdots \cap U_{i_p})$$

where the product on the right is over all sequences $(i_0, ..., i_p)$ of length $p + 1$.

We denote by $C_{\text{Cech}}(\mathfrak{U}, F)$ the family of all $C^p_{\text{Cech}}(\mathfrak{U}, F)$ for $p = 0, 1, ...$



Intuitively, the Cech cohomology of the covering $\mathfrak{U}$ with values in a sheaf F should be the simplicial cohomology on the nerve $\mathcal{N}(\mathfrak{U})$ where one makes the obvious replacement of "functions" f with variable codomain in place of functions with a fixed codomain. In particular, the boundary map in the simplicial complex induces a coboundary map between the Cech cochain groups defined by the following rule:

$$\begin{aligned} \delta : C^p_{Cech}(\mathfrak{U}, F) &\to C^{p+1}_{Cech}(\mathfrak{U}, F) \\ f &\mapsto \sum_{j=0}^{p+1} (-1)^j \rho_j (f_{i_0 \dots \hat{i}_j \dots i_{p+1}}) \end{aligned}$$

where ρ_j denotes the restriction homomorphism

$$\rho_j : F(U_{i_0 \dots \hat{i}_j \dots i_{p+1}}) \to F(U_{i_0 \dots i_{p+1}}).$$

What is really going on? Ultimately, it comes down to the restriction homomorphisms. Each $(p+1)$-simplex $\{i_0, ..., i_{p+1}\}$ has $p+2$ 'faces' $\{i_0 ... \hat{i}_j ... i_{p+1}\}$ obtained by deleting one of the coordinates i_j. Each face has support $U_{i_0 \dots \hat{i}_j \dots i_{p+1}}$, whose union $U_{i_0 \dots i_{p+1}}$ is the entire support of $\{i_0, ..., i_{p+1}\}$, and therefore any element of $F(U_{i_0 \dots \hat{i}_j \dots i_{p+1}})$ can be restricted to an element of $F(U_{i_0 \dots i_{p+1}})$.

Since $\partial \circ \partial = 0$, we have $\delta \circ \delta = 0$. Thus δ makes $C_{\text{Cech}}(\mathfrak{U}, F)$ into a cochain complex called the *Cech cochain complex*. We call the homology of this cochain complex its *Cech cohomology*

$$H^p_{Cech}(\mathfrak{U}, F) = \frac{\ker \delta^p}{\operatorname{im} \delta^{p-1}}.$$

Example 4.29. The following is a very typical example (e.g. Example 4.0.4 in Hartshorne).

Let S^1 be the circle with its usual topology, and let $\mathbb{Z}_{S^1}$ be the constant sheaf with value $\mathbb{Z}$. Let $\mathfrak{U}$ be the open covering by two connected open semi-circles U, V which overlap at each end.

Then

$$C^0_{Cech}(\mathfrak{U}, \mathbb{Z}_{S^1}) = \mathbb{Z}_{S^1}(U) \times \mathbb{Z}_{S^1}(U) = \mathbb{Z} \times \mathbb{Z}$$

$$C^1_{Cech} = \mathbb{Z}_{S^1}(U \cap V) = \mathbb{Z} \times \mathbb{Z}.$$

Note that $\mathbb{Z}_{S^1}$ already "knows" the fact that $U \cap V$ is disconnected in the second equality.

The map $\delta_0 : C^0_{Cech} \to C^1_{Cech}$ takes (a, b) to $(a - b, a - b)$. Then

$$H^0_{Cech}(\mathfrak{U}, \mathbb{Z}) = \ker \delta_0 = \{(a, a) \in \mathbb{Z} \times \mathbb{Z} \simeq \mathbb{Z},$$

and

$$H^1_{Cech}(\mathfrak{U}, \mathbb{Z}) = \frac{C^1}{\text{im } \delta_0} = \frac{\mathbb{Z} \times \mathbb{Z}}{\{(a-b, a-b) \in \mathbb{Z} \times \mathbb{Z}\}} \simeq \frac{\mathbb{Z} \times \mathbb{Z}}{\mathbb{Z}} \simeq \mathbb{Z},$$

as expected.

Proposition 4.30. $H^0_{Cech}(\mathfrak{U}, F) = F(X)$.

Proof. A 0-cochain is a system of sections $(f_i)_{i \in I}$ with every f_i being a section of $F(U_i)$. It is a cocycle if and only if it satisfies $f_i - f_j = 0$ over $U_i \cap U_j$, or in other words if there is a section f of F on X coinciding with f_i on U_i for all $i \in I$. □

Based on the proposition above, we can directly define the *0-th sheaf cohomology group* $H^0_{\mathbf{Sh}}(X, F)$ of a topological space X in terms of the global sections functor on X, since $H^0_{Cech}(\mathfrak{U}, F) = F(X)$ is independent of the choice of open covering $\mathfrak{U}$ of X. For the higher cohomology of X, this is not always true.

To define the sheaf cohomology of X more generally, we have two options:

1. by a theorem of Serre, $H^p_{Cech}(\mathfrak{U}, F) \simeq H^p_{\mathbf{Sh}}(X, F)$ on a sufficiently nice space X (paracompact and Hausdorff) whenever $\mathfrak{U}$ is a good cover (all open sets and finite intersections of open sets are contractible). Unfortunately, the Zariski topology is not Haudorff.

2. More generally, define $H^p(X, F)$ as the colimit of groups $H^p(\mathfrak{U}, F)$, where the colimit is taken over finer and finer refinements of coverings of X.

Following Serre, we will take the second approach.

Definition 4.31. A covering $\mathfrak{U} = \{U_i\}_{i \in I}$ is said to be *finer* than the covering $\mathfrak{V} = \{V_j\}_{j \in J}$, denoted $\mathfrak{U} \preceq \mathfrak{V}$, if there exists mapping $\tau : I \to J$ such that $U_i \subset V_{\tau(i)}$ for all $i \in I$. This defines a preorder between coverings of X.

Lemma 4.32. *The finer-than relation $\mathfrak{U} \preceq \mathfrak{V}$ defines a directed preorder on the set of coverings of X.*

Suppose $\mathfrak{U} \preceq \mathfrak{V}$. If f is a p-cochain, put

$$(\tau f)_{i_0, \ldots, i_p} = \rho^V_U(f_{\tau(i_0) \ldots \tau(i_p)})$$

where ρ^V_U denotes the restriction homomorphism defined by the inclusion of $U_{i_0 \ldots i_p}$ in $V_{\tau(i_0) \ldots \tau(i_p)}$. The mapping $f \mapsto \tau f$ is a homomorphism from $C^p_{Cech}(\mathfrak{V}, F)$

to $C^p(\mathfrak{U}, F)$ defined for all $p \geq 0$ and commuting with δ, thus it also defines a homomorphism on homology

$$\sigma : H^p_{Cech}(\mathfrak{V}, F) \to H^p_{Cech}(\mathfrak{U}, F)$$

By a proposition of Serre, we know that the homomorphism σ depends only on $\mathfrak{U}$ and $\mathfrak{V}$, not on the choice of τ.

Definition 4.33. The *p-th cohomology group of X with values in a sheaf F* is given by

$$H^p_{\mathbf{Sh}}(X, F) = \operatorname*{colim}_{\sigma} \ H^p_{Cech}(\mathfrak{U}, F)$$

where the colimit is defined over (the directed system of maps σ generated by) the directed preorder of coverings $\mathfrak{U}$ of X.

We have defined the sheaf cohomology groups of X, but we are far from done—we need to check that various tools like long exact sequences, connecting morphisms, and homotopy invariance actually exist for $H^q_{\mathbf{Sh}}(X, F)$, turning the functor H into a full-fledged cohomology theory.[27] Nor have we seen the very elegant application of sheaf cohomology to the familiar cohomology of manifolds, e.g. de Rham and singular cohomology (the upshot: there's an incredible amount of structure hidden in just the constant sheaf $\mathbb{R}_X$ over a smooth manifold). We will return to these aspects later.

4.4 Coherent algebraic sheaves

Coherent algebraic sheaves take the role of finite-dimensional vector spaces in sheaf cohomology. In this section, I will give the definitions for Serre's sheaf of relations, coherent sheaves, and algebraic sheaves, and conclude with the main result for affine varieties.

First, some preliminaries.

1. For a continuous map $\psi : X \to Y$ and F a sheaf on X, the *direct image sheaf* $\psi_* F$ is a sheaf on Y given by $\psi_* F(U) = F(\psi^{-1}(U))$.

2. Given a two sheaves F, G on X with values in the same category, a *morphism of sheaves* $\phi : F \to G$ is just a natural transformation of presheaves that is compatible with the restriction maps. In other words, the following diagram commutes:

$$\begin{tikzcd} F(U) \arrow[r, "\phi_U"] \arrow[d, "r_{V,U}"'] & G(U) \arrow[d, "r_{V,U}"] \\ F(V) \arrow[r, "\phi_U"] & G(V) \end{tikzcd}$$

3. A *morphism of ringed spaces* $f : (X, F) \to (Y, G)$ is a continuous map $f : X \to Y$ together with a morphism of sheaves, $f^* : G \to f_* F$.

[27] In practice, people do not usually prove these properties directly, since they drop out almost automatically from the derived functor definition of (sheaf) cohomology.

Definition 4.34. Let F be a sheaf of A-modules, and let $s_1, ..., s_p$ be sections of F over an open $U \subset X$. If we assign to any family of germs $f_1, ..., f_p$ in A_x the element

$$\sum_{i=1}^{p} f_i \cdot s_i(x) \in F_x$$

we obtain a homomorphism $\phi : A^p \to F$ defined over U (being precise, ϕ is a fixed homomorphism from $A^p(U)$ to $F(U)$). The kernel $\ker \phi = R(s_1, ..., s_p)$ is a subsheaf of A^p called the *sheaf of relations* between the s_i.

A sheaf of relations $R(s_1, ..., s_p)$ is essentially the sheaf-theoretic version of $\mathbb{V}(I)$, i.e. an operation that cuts out an affine variety using some algebraic data.

The following definition is adapted from Serre:

Definition 4.35. A sheaf F of A-modules over X is *coherent* if it is:

1. *finitely generated* or of *finite type*, i.e. each $F(U)$ is generated (under multiplication by elements of $A(U)$) by a finite number of sections,

2. if $s_1, ..., s_p$ are sections of F over an open subset $U \subset X$, then the sheaf of relations between the s_i, restricted to U, is also of finite type

Roughly, the first condition corresponds to the fact that X has some finite covering by affine opens, while the second condition corresponds to the requirement that, over any open set U, the set of polynomials that vanish over U is finitely-generated (i.e. every affine variety is cut out by finitely-many polynomials). The dimension over k of $F(U)$ is exactly the number of polynomials needed to generate U.

By comparison, Hartshorne defines a coherent sheaf as a quasi-coherent sheaf that is "finitely presented" with respect to a cover. Unfortunately, the definition in Hartshorne is quite convoluted and hard to work with (perhaps because he was defining it in the more general case, for schemes), so let me give the nLab's version: a *quasi-coherent sheaf* is a sheaf of A-modules that is locally presentable on some cover $\{U_i\}$, in the sense that for every i there exists an exact sequence of sheaves

$$A^{I_i}|_{U_i} \xrightarrow{T} A^{J_i}|_{U_i} \to F|_{U_i} \to 0.$$

The middle arrow is surjective, so we can think of each $F|_{U_i}(U_j)$ as the cokernel of T, i.e. a map between free modules. I like to think in terms of vector spaces, so I imagine $F|_{U_i}(U_j)$ as the cokernel of a linear transformation T between two fixed vector spaces. Note that these are fixed by i, not by j. So $F|_{U_i}$ really represents a class of vector spaces, all of which are fixed by two "dimensionality" parameters: the cardinality of I_i and J_i. When I_i and J_i are finite, we say that the sheaf is coherent. So coherent sheaves are, roughly, analogous to finite-dimensional vector spaces.[28]

[28] Several authors claim that one can very well use quasi-coherent sheaves in place of coherent sheaves even in the sheaf cohomology of affine varieties, but Serre does not develop this fact in FAC.

Of course, in algebraic topology we don't always or even usually take our (co)homology coefficients in a field, so there's no special reason to expect our "coefficients in a sheaf" to look like finite-dimensional vector spaces. And indeed, the situation is generalized when we move to schemes defined over arbitrary rings.

Finally:

Definition 4.36. Suppose X is an affine variety with its sheaf of local rings $\mathcal{O}_X$ (so a ringed space). We call an *algebraic sheaf* on X any sheaf of $\mathcal{O}_X$-modules.

An algebraic sheaf F over X is said to be *coherent* if it is a coherent sheaf of $\mathcal{O}_X$-modules. In particular, the cohomology groups of X with values in a coherent algebraic sheaf F will be finite-dimensional vector spaces over k.

Proposition 4.37. *The sheaf of regular functions $\mathcal{O}_X$ on an affine variety X over k is a coherent algebraic sheaf.*

Proof. Essentially, consider $\mathcal{O}_X$ as a sheaf of modules over itself, and note that the module of "relations between polynomials", with elements of the form $g_i f_i = 0$, is finitely generated, since the underlying polynomial ring is Noetherian. The full proof is in FAC, §37. □

Proposition 4.38. *Let X be an affine variety, $\{q_i\}$ a family of regular functions on X that do not vanish simultaneously, and $\mathfrak{U}$ the open covering of X consisting of $X_{q_i} = U_i$. If F is a coherent algebraic subsheaf of $\mathcal{O}_X^p$, then $H^k_{Cech}(\mathfrak{U}, F) = 0$ for all $k > 0$.*

Intuitively, this makes sense. If we're covering affine varieties by other affine opens, then we would expect that the only obstruction would be at the level of connected components, i.e. in $H^0_{Cech}(\mathfrak{U}, F)$.[29] The proposition gestures toward the following fact: a (quasi-)projective variety is really just a gluing of local affine pieces. The $\mathcal{O}_X$-module structure both tracks and linearizes the data of these pieces on overlaps. We expect that, when restricting to the affine pieces, to find only trivial higher cohomology, since they are the elements of our "good covering", and the proposition above verifies our intuition. To see a proof of Serre duality, we need to apply sheaf cohomology to projective and abstract varieties.

4.5 Serre duality

An *algebraic variety over k* or an *abstract k-variety in the sense of Serre* is a topological space X and a sheaf $\mathcal{O}$ of germs of mappings from X to k, which is additionally

[29] The truly striking thing about the proposition is not its proof, which is straightforward once all of the machinery is set up, but the very idea that one could apply coherent sheaves, which had been used to work with the topology of complex manifolds, to the relative paucity of the Zariski topology on affine varieties. Nowadays we take it for granted that sheaves work on the Zariski topology; earlier, I referenced sheaf theory as a tool designed to work on such topologies.

1. *a prealgebraic variety*, i.e. X has a finite open covering $\mathfrak{U} = \{U_i\}_{i \in I}$ such that each $(U_i, \mathcal{O})$ is isomorphic (via a morphism of ringed spaces) to an affine variety V_i with its structure sheaf.

2. *separated*, i.e. the image of the diagonal $X \to X \times X$ is closed in $X \times X$.

Example 4.39. Both affine varieties and projective varieties are examples of algebraic varieties. In particular, the projective space $\mathbb{P}_r(k)$ of dimension r over k is an algebraic variety. This is important, since sheaf cohomology on any projective algebraic variety can be reduced to sheaf cohomology on $\mathbb{P}_r(k)$.

The fact that k is a field makes (pre)algebraic varieties into ringed spaces. The first condition is the ringed-space analogue to the usual way we define the Zariski topology and structure sheaf. The second condition is analogous to the Hausdorff (T2) separation axiom [110, pg. 68] and guarantees our ability to glue subvarieties of X into new varieties.

While I will review a proof of Serre duality (for curves) here, following FAC, [97], and [114], I would also like to illustrate the following point: the ringed space, sheaf-theoretic definition of algebraic varieties "comes with" the appropriate notion of cohomology as embedded in the definition of the structure sheaf.

First, we observe the behavior of (non-coherent) sheaf cohomology on curves. The following is Proposition 4 in §53 of FAC.

Proposition 4.40. *If C is an irreducible algebraic curve and $\mathcal{F}$ is an arbitrary sheaf in C, we have $H^n(C, \mathcal{F}) = 0$ for $n \geq 2$.*

Proof. The proof leans heavily on the fact that the closed subsets of C which are not C itself are *finite*. For a finite subset S and a point $x \in S$, we define $U_x = (C - S) \cup \{x\}$; the family $\{U_x\}_{x \in S}$ forms an finite open covering $\mathfrak{U}_S$ of C. We will take the following lemma as a fact: coverings of the type $\mathfrak{U}_S$ can be made arbitrarily fine, by varying S.

Now take any sheaf F, and set $W = C - S$. It is clear that $U_{x_0} \cap ... \cap U_{x_n} = W$ for distinct x_i if $n \geq 1$. If we put $G = \mathcal{F}(W)$, it follows that the alternating complex $C'(\mathfrak{U}_S, \mathcal{F})$ is isomorphic, in dimensions ≥ 1, to $C'(\Delta(S), G)$, where $\Delta(S)$ denotes the simplex with S for its set of vertices. Since $H^n(\Delta(S), \mathcal{F}) = 0$ by Proposition 4.38, it follows that $H^n_{Cech}(\mathfrak{V}^S, \mathcal{F}) = H^n(\Delta(S), G) = 0$ for $n \geq 2$. □

The analogous vanishing theorem for n-dimensional varieties was proved by Grothendieck some years after FAC. Since we will only develop, for now, in the case of Serre duality for curves, we will not need those results.

The following version is adapted from [114] (itself adapted from [97]).

Theorem 4.41 (Serre duality for curves)**.** *Let C be a non-singular projective curve, let $\mathcal{L}$ be an invertible sheaf with dual $\mathcal{L}^\vee$, and let $\Omega^1_C = \Omega^1$ be the invertible sheaf of differential forms on C. There is a natural perfect pairing $H^0(C, \Omega^1 \otimes \mathcal{L}^\vee) \times H^1(C, \mathcal{L}) \to \bar{k}$. Hence $h^1(C, \mathcal{L}) = h^0(C, \Omega^1 \otimes \mathcal{L}^\vee)$.*

Assuming Serre duality, the proof of Riemann-Roch then proceeds as follows. Recall that genus g of C is defined as $g = h^0(C, \Omega^1)$.

$$\begin{aligned}
h^0(C,\mathcal{L}) - h^0(C,\Omega^1 \otimes \mathcal{L}^\vee) &= h^0(C,\mathcal{L}) - h^1(C,\mathcal{L}) && \text{(Serre duality)} \\
&= \chi(C,\mathcal{L}) \\
&= d + \chi(C,\mathcal{O}_C) && (\mathcal{L} \text{ invertible of deg } d) \\
&= d + h^0(C,\mathcal{O}_C) - h^1(C,\mathcal{O}_C) \\
&= d + 1 - h^0(C,\Omega^1) && \text{(``analytic facts'')} \\
&= d + 1 - g.
\end{aligned}$$

Proof of Serre duality. We begin by noting some facts about Serre's *twisted sheaf* $\mathcal{O}_C(m) = \mathcal{O}(m)$, a basic construction for talking about homogeneous polynomials (i.e. the set of global sections of $\mathcal{O}(m)$ on projective space is precisely the vector space of homogeneous polynomials of degree m; an element of the stalk $\mathcal{O}(m)_x$ is a rational function P/Q homogeneous polynomials, with $\deg P - \deg q = m$).

First, for a given algebraic sheaf $\mathcal{F}$ defined on projective space $\mathbb{P}^r(k)$, the twisted sheaf $\mathcal{F}(m)$ is defined by $\mathcal{F}(m) := \mathcal{F} \otimes_{\mathcal{O}} \mathcal{O}(n)$. (There is a more involved, elementary definition in terms of gluings of sheaves, but this is sufficient for our purposes.)

Second, every twisted sheaf is invertible. Recall that a sheaf $\mathcal{L}$ is *invertible* if it is a coherent sheaf of $\mathcal{O}_C$-modules for which there is an inverse sheaf $\mathcal{T}$ such that $\mathcal{L} \otimes \mathcal{T} = 1$, the monoidal unit. Invertible sheaves are the sheaf-theoretic version of line bundles. Recalling our discussion of Riemann-Roch, an invertible sheaf $\mathcal{L}$ corresponds to the old notion of divisor, and the degree d of an invertible sheaf corresponds to the degree of a divisor. It is calculated by $\deg(\mathcal{L}) = \chi(C,\mathcal{L}) - \chi(C,\mathcal{O}_C)$, where $\chi(X,\mathcal{F}) = \sum(-1)^i \dim_k H^i(X,\mathcal{F})$ is (homological) Euler characteristic.

Third, every invertible sheaf $\mathcal{L}$ is of the form $\mathcal{O}_C(p_1 + ... + p_a - q_1 - ... - q_b) = \mathcal{O}(D)$ for some divisor D, where $\deg \mathcal{L} = a - b$. This result is from [114], but I can't find a proof anywhere.

Lastly, any coherent algebraic sheaf $\mathcal{F}$ is generated by global sections after enough "twists". (This is Theorem 2 in §66 of FAC, which Serre proves by taking a long detour through the theory of graded R-modules of finite type.) That is, there is some number m_0 such that for all $m \geq m_0$, the twisted sheaf $\mathcal{F}(m)$ is generated by a finite number of global sections.

Let D be a divisor on C, and consider the sheaf $\mathcal{O}(D)$. We will first reinterpret $I(D) := H^1(C,\mathcal{O}(D))$ in the language of *repartitions* (also known as *adeles*). A *repartition* R is an indexed set $\{r_P\}_{P \in C}$ where r_P is an element of the function field $k(C)$, and $r_P \in \mathcal{O}_P(P)$ for all but finitely many P. Note that R is a $k(C)$-algebra.

Then $I(D) \simeq \frac{R}{R(D)+k(C)}$. Missing parts: (1) complete the proof of Serre duality; I am having some trouble understanding the later parts of the proof in [114], in terms of repartitions. I'll try looking through Serre's proof in [97]. (2) additional arguments as to why we should think of ringed spaces as "coming with a cohomology theory". (3) additional review/example of deriving the long exact sequences for sheaves.

□

Add here: additional discussion of the difference / connection between topological and algebraic invariants. What Riemann-Roch sets up later, e.g.... How Toen (in lecture) notes the distinction between topological and algebraic invariants on the two sides of the Hodge decomposition in H-R-R:

$$H^i(X, \mathbb{Q}) \otimes \mathbb{C} \simeq \bigoplus_{p+q=i} H^i(X, \Omega^q_X),$$

as well as on the two sides of the étale ℓ-adic cohomology, $H^i_{et}(X, \mathbb{Q}_\ell) \simeq$ sheaf cohomology for the Zariski topology.

Maybe hint at non-commutative geometry by giving the definition of a non-commutative variety:

Definition 4.42. A *non-commutative variety over some base commutative ring* k is a k-linear (dg-)category.

For example, if A is a k-algebra, then $D(A)$ is the dg-category of complexes of A-modules.

For example, schemes over k, which is the same thing as above where one adds the sheaf gluing.

The point of all this is, and why it might be important for us, is that even without geometric things like open sets, topology, we can still define algebraic things like differential forms (via Hochschild homology) and topological things like ℓ-adic cohomology using (co)homology.

4.6 Good cohomology, part 2

Using sheaf cohomology, one can extend Riemann-Roch to non-singular projective varieties of arbitrary dimension, over an arbitrary field k. But just saying that sheaf cohomology allowed one to generalize Riemann-Roch doesn't quite capture what algebraic geometers mean by "a good cohomology theory for abstract varieties", nor does it do justice to Serre's motivation for developing sheaf cohomology. There are three interpretations of good cohomology, which I've labeled as "the one from topology", "the one from number theory", and "the one from scheme theory", the last of which is really a synthesis of the interpretations from topology and number theory.

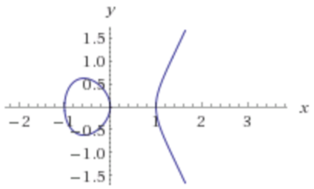
From topology. Sheaf cohomology, in the first place, is modeled on classical cohomology theories in topology. So there should be a good way to handle "algebraic cycles"—higher co-dimension generalizations of divisors—as if they were cycles in the topological sense. There should be a good notion of what gluing two varieties does to their cohomology, e.g. something like excision and Mayer-Vietoris as in the singular case. Developments down this line include Grothendieck's étale cohomology, which uses the topology of the étale space to define coherent analogues of genus for schemes, Sullivan's elaboration of localization for geometric topology, and Voevodsky and Morel's $\mathbb{A}^1$-homotopy theory for schemes, which begins by trying to make the affine line $\mathbb{A}^1$ behave more like the topological interval used in the definition of homotopy. In general, "good" cohomology is an organizing principle that seeks to make the rigid geometry of polynomial functions softer and more topological.

From number theory. From another direction entirely came the remarkable observations by F. K. Schmidt and Weil relating Riemann-Roch to number theory, e.g. to "abstract Riemann surfaces" defined over finite fields. The Weil conjectures, which Weil developed to solidify the link between number theory to algebraic geometry, motivated much of the work by Serre and Grothendieck from 1955-1960. They state that certain, extremely hard facts about number fields—e.g. regularity results about nonlinear phenomena like the zeta function $\zeta(s)$—can be decomposed and "linearized", via Weil cohomology, in terms of the function field of the appropriate finite field. Good cohomology, in the sense of Weil cohomology, was formulated as precisely the kind of cohomology theory that would provide a witness to and a resolution of the Weil conjectures.

I will come back to the Weil conjectures in Section 4.7.

From scheme theory. The view from topology, while giving the right motivation, abstracts over certain basic, geometric facts about polynomials, facts which (with the benefit of hindsight) were necessary for the development of a number-theoretic interpretation of polynomials or, more accurately, a polynomial-theoretic interpretation of numbers. Scheme theory lifts those geometric facts to the level of cohomology, in the process offering a entirely new perspective of what a polynomial function "really" is.

Classically, a polynomial function is like the long handle of a sledgehammer—rigid, easy-to-grasp, and attached to a blunt but powerful tool: the geometric shape defined by its null set. The Nullstellensatz lets us grasp and swing the hammer as an object in its own right, rather than always and awkwardly as the combination of a handle and a hammer head. It tells us that, in the affine case, isomorphism of coordinate rings (counted by equalities of polynomials) corresponds to equality of shapes. For example, the affine algebraic curve defined by $y^2 = x^3 - x$ is a geometric shape



with two coordinate functions: y, which returns the y value of any point on the curve, and x, which returns the x value of any point on the curve. This generates the coordinate ring of $y^2 = x^3 - x$, under the constraint that functions generated by the coordinate functions, e.g. $y^2 + x$ and x^3, are identified when they are equal on the curve. The structure of the coordinate ring determines the shape of the variety. Algebraic properties of the coordinate ring correspond directly to geometric properties of the shape; for instance, the ring of coordinate functions has unique factorization precisely when every line bundle over the shape looks the same as every other.

In light of scheme theory, sheaves became useful in algebraic geometry because the Nullstellensatz was incapable of distinguishing singular varieties over arbitrary rings or even over $\mathbb{Q}$. Recall that the Nullstellensatz for affine varieties can be condensed into the following set of (order-reversing) bijections:

$$\begin{aligned}\text{affine varieties in } k^n &\leftrightarrow \text{radical ideals in } R = k[x_1, ..., x_n]\\ \text{irreducible varieties in } k^n &\leftrightarrow \text{prime ideals in } R\\ \text{points in } k^n &\leftrightarrow \text{maximal ideals in } R\end{aligned}$$

where, importantly, k is an algebraically-closed field. As we know, ideals overgenerate varieties even when k is algebraically closed: for example, the ideals (x^2) and (x) both specify the same variety in $\mathbb{C}$. The Nullstellensatz solved this problem by passing to radical ideals; different radical ideals specify different varieties. Unfortunately, passing from an ideal to its radical creates its own problems. In particular, it does not preserve the local information about multiplicities at potential singularities, e.g. (x) has a 0 of multiplicity 1 while (x^2) has a 0 of multiplicity 2 at the origin.

What happens if k is not algebraically closed? Take $k = \mathbb{R}$. Then $\mathbb{V}(I) = \emptyset$ for some $I \subsetneq k[x_1, ..., x_n]$, in that we have situations like $\mathbb{I}(\mathbb{V}(x^2 + 1)) = (1) \neq (x^2 + 1)$. So the bijection between varieties over k^n and radical ideals in $k[x_1, ..., x_n]$ fails in $\mathbb{R}$; in fact, each bijection above fails. More basically, when k is not algebraically closed, or when k is a ring, we will not see or distinguish all the possible zeroes (counting multiplicity). This can be seen in Bezout's theorem: two generic plane algebraic curves in projective space have intersection of size at most the product of their degrees (counting multiplicity and intersections at infinity), with equality if they are defined over an algebraically closed field. What is missing from the picture over $\mathbb{R}$ or some other number field is the information about the multiplicity, i.e. the topology at that point.

Enter sheaves. The first real evidence that we need sheaves of rings instead of just rings arises in the need to record these multiplicities, i.e. in the need to record which functions are *not* regular (i.e. blow-up) at x. In scheme theory, this accounting is done automatically when one localizes over ideals $\sim$ Zariski-open sets in $\operatorname{Spec} A$. The machinery of sheaf cohomology then churns this data back into a topological invariant, e.g. generalizations of genus for varieties over fields that are not $\mathbb{C}$.

To go back to the hammer analogy: the sheaf is an additional piece of data that measures the impact of the hammer on a given "test surface", represented by the particular field or ring k. It is, in fact, exactly the *geometric* data, that says X is not just any topological space but is in fact a variety. I like to think of a sheaf as something broadly similar to the Nullstellensatz: as the series of adjustments that one makes when handling a hammer, as a way of adapting to its weight and the way it strikes the surface.

4.7 The Weil conjectures

In Question 17, I offered a hypothesis of what sheaves are for—cohomology—and asked for an explanation of that hypothesis in the language of classical algebraic geometry. In Section 4.2, I formed an explanation that began with the classical versions of Riemann-Roch and genus, and then showed how sheaf cohomology facilitated extensions of Riemann-Roch to arbitrary fields (still algebraically-closed). In Sections 4.3-4.5, I reviewed the cohomological machinery needed to prove this result, up to Serre duality. In Section 4.6, I then stated that a good cohomology theory (in algebraic geometry) was something that imitated cohomology theories in algebraic topology, and that this meant, for the purposes of number theory, that it ought to be a cohomological witness to the Weil conjectures. In this last section, I explain what it means for a cohomology theory to witness the Weil conjectures, and how the Weil conjectures relate back to my characterization of category theory as a "computational" lens on algebraic geometry.

The goal of this section is not to review the Weil conjectures and Weil cohomology in full, which is done in far better detail in [87], [23], and in the appendix of Hartshrone. The goal to answer the following question:

Question 18. Why should we care about solutions $\mathbb{V}(I)$ over $\mathbb{Q}$ and $\mathbb{F}_q$ if all the information already lives in $\mathbb{C}$? Why does it matter that we can change the base field, much less that we can change it to something finite?

If the import of Question 18 is not clear from Galois theory, consider that it is also closely related to the following question:

Question 19. Why did Weil, Serre, and Grothendieck seek an abstract (and eventually sheaf-theoretic) definition of "algebraic variety" in the first place?

In a straightforward and possibly naive way, the answer to both Question 19 and Question 18 is "because number theory".[30] But I would like to also consider a different (and still possibly naive) answer to Question 18: changing the base field matters *because* of Question 19. But first I will review the straightforward answer.

The Weil conjectures have their origins in Riemann's study of the zeta function

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

[30] Question 18 is related to a theme in algebraic geometry which Dieudonne called "extending the scalars: [...] the introduction of complex points and later of generic points were the forerunners of what we now consider as perhaps the most characteristic feature of algebraic geometry, the general idea of change of basis" [27, pg. 828]. Similarly, Question 19 is related to a theme which Dieudonne called "Extending the space: [...] projective geometry and n-dimensional geometry paved the way for the modern concepts of "abstract" varieties and schemes."

over $s \in \mathbb{C}$. Consider that we can think of $\mathbb{C}$ as being obtained from $\mathbb{R}$ by adjoining the square root of -1, i.e. by adding the roots of the equation x^2+1. Just so, the underlying inspiration for Weil's conjectures is the relation between *number fields*, which one can obtain by adjoining solutions of more general equations to $\mathbb{Q}$, and *function fields over finite fields*. The importance of this relation cannot be overstated. If we had to sum up the passage from the Nullstellensatz to FAC to SGA and even to Deligne's proof of the final Weil conjecture, it would be something like "how to think about the discrete, e.g. a finite field $\mathbb{F}_q$, in terms of the continuous, e.g. the topology of the field of rational functions over $\mathbb{F}_q$". For reasons that are too complicated to explain here—I will merely point to Dieudonne's excellent history of algebraic geometry [27]—the geometry of the 20th century had turned away from ways of computing the solutions to polynomial equations, which had motivated the development of algebraic geometry up to the time of Galois. Beginning at around the time of Riemann-Roch, the fact that all the (algebraic) solutions lived in $\mathbb{C}$, or that taking coefficients over $\mathbb{C}$ preserved all the (geometric) information about multiplicity, became less important than understanding and organizing the constellation of invariants above any algebraic variety, and it turned out that the choice of base field k played an enormous role.

Suppose that X is a non-singular, n-dimensional projective algebraic variety over the finite field $\mathbb{F}_q$. The zeta function $\zeta_X(s)$ of X is by definition

$$\zeta_X(s) = \exp\left(\sum_{m=1}^{\infty} \frac{N_m}{m} q^{-ms}\right)$$

where N_m is the number of points of X defined over the degree m extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$. Then the Weil conjectures are:

1. (Rationality) $\zeta_X(s)$ is a rational function of s.

2. (Riemann hypothesis) More precisely, if $n = \dim X$, $\zeta_X(s)$ can be written as a finite alternating product

$$\zeta_X(s) = \frac{P_1(s)}{P_0(s)} \frac{\cdots}{\cdots} \frac{P_{2n-1}(s)}{P_{2n}(s)}$$

   where each root of each $P_k(s)$ is a complex number of norm $q^{-k/2}$. This implies that all zeros of $P_k(s)$ lie on the "critical line" of complex numbers s with real part $k/2$.

3. (Functional equation) The roots of $P_k(s)$ are interchanged with the roots of $P_{2n-k}(s)$ under the substitution $s \mapsto \frac{1}{q^n s}$.

4. (Betti numbers) If X is a "reduction mod p" of a non-singular projective variety $\tilde{X}$ defined over a number field embedded in $\mathbb{C}$, then the degree of P_k is the k-th Betti number of $\tilde{X}$ with its usual topology.

The form of the conjectures stated above is slightly modified from that of [87].

The last, topological ingredient came with the definition of Weil cohomology. Weil observed that the number of points of X over $\mathbb{F}_{q^m}$ (i.e. the number of solutions of X in that field) is equal to the number of fixed points of the Frobenius automorphism $\phi_{q^m} : \bar{X} \to \bar{X}$, $x_i \mapsto x_i^{q^m}$, where $\bar{X}$ indicates the lift of X over its algebraic closure. In algebraic topology, the Lefschetz fixed-point theorem states that this second number can be calculated as an alternating sum of the traces of maps induced by ϕ_{q^m} on the cohomology groups of X. This motivates the following definition:

Definition 4.43. A *Weil cohomology theory* is a cohomology theory for non-singular projective varieties (over *any* field k, but particularly finite fields) satisfying Poincaré duality and some form of the Lefschetz fixed-point theorem.

Example 4.44. Let k be the base field. If $\operatorname{char}(k) = 0$, we have *algebraic de Rham cohomology*, with coefficient field k itself. If $k = \mathbb{C}$, then we have the standard de Rham cohomology in terms of differential forms.

Example 4.45. Let k be the base field. If $\sigma : k \to \mathbb{C}$ is an embedding of k into the field of complex numbers, we have the so-called *Betti cohomology* associated to σ, which is just the singular cohomology of the variety viewed as a complex variety by means of the embedding σ. The singular cohomology here is taken with rational coefficients, so the coefficient field of Betti cohomology is the field $\mathbb{Q}$ of rational numbers.

Example 4.46. Let k be the base field. For ℓ a prime number different from the characteristic of k, we have *ℓ-adic cohomology*, also known as ℓ-adic étale cohomology. The coefficient field is the field $\mathbb{Q}_\ell$ of ℓ-adic numbers. The ℓ-adic cohomology groups are vector spaces over this field.

Suppose H^* is a Weil cohomology theory. Then it is defined on X, and we can calculate the number of fixed points of applying ϕ_{q^m} to X in terms of H^*, where each term $P_k(s)$ in $\zeta_X(s)$ corresponds to the induced Frobenius action on $H^k(X)$. Almost immediately, we can verify rationality, the Riemann hypothesis, and the Betti numbers for X. The functional equation follows from Poincaré duality.

So the Weil conjectures clearly motivate the search for solutions over $\mathbb{F}_q$ “because number theory”.

A brief diversion

In 1994, Karl Sims presented a paper at SIGGRAPH called “Evolving Virtual Creatures” [99]. He simulated a range of different block-like creatures, and competed them against each other in activities like swimming, walking, and jumping. The creatures have a genotype as well as a phenotype, and the “genes” of the winner are passed on, via a genetic algorithm, to a second generation of creatures.

Figure 4.2: Animated swimming creatures from [99].

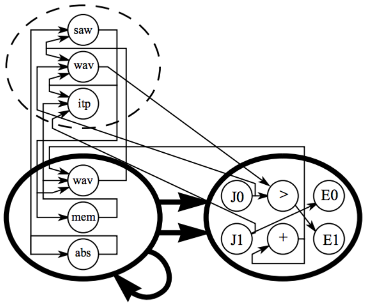


Figure 4.3: Genotype governs both the morphology and the control structure.

The interesting thing about the paper is that Sims *co-evolved* the morphology of his creatures in tandem with their control structures. "When a creature is synthesized from its genetic description, the neural components described within each part are generated along with the morphological structure. This causes blocks of neural control circuitry to be replicated along with each instanced part, so each duplicated segment or appendage of a creature can have a similar but independent local control system. These local control systems can be connected to enable the possibility of coordinated control. [...] In this way the genetic language for morphology and control is merged. A local control system is described for each type of part, and these are copied and connected into the hierarchy of the creatures body to make a complete distributed nervous system."

The Weil conjectures, redux

At the beginning of this note, I asked in Question 16 why category theory proved so useful in algebraic geometry, secretly knowing it was accepted as useful only with Grothendieck and Deligne's proof of the Weil conjectures. I did not dispute the power of category theory to describe many facets of algebraic geometry—I simply wanted to know *why* we would want to do so. I then claimed that category theory's success in algebraic geometry arose because it was a *computational** lens on algebraic geometry, and that this success owns much to sheaf cohomology, which allows us to manipulate the "accounting" aspects of algebraic geometry *in tandem* with its topological axiomatization. What makes the particular, *sheaf* cohomology theories of algebraic geometry computational* rather than merely / overwhelmingly technical is the fact that the Zariski topology on affine and projective space is in some sense *artificial*—it is an axiomatization of something else entirely from the natural or synthetic topology defined on k.[31] This is not simply an extra parameter that needs to be tracked through

[31] But then again, "natural topology" only makes sense for certain k.

the process of constructing a cohomological invariant, but a whole sequence of external data, relations, and syzygies that sits "outside the category", with its own internal language and "theory of computation", in a sense yet to be made precise.

To be clear, manipulating the accounting and the topological axiomatization is a good thing for much the same reason that functoriality is a good thing—it makes things work. Manipulating the accounting comes down to (categorical) operations on sheaves we have already seen: morphisms of sheaves, short and long exact sequences, direct images, and so on. As for the axiomatization: we have not yet seen any other topological axiomatization aside from the Zariski topology, since we have developed sheaf cohomology only up to end of FAC. But in fact, later axiomatizations such as the étale topology and the Nisnevich topology would be used to address defects in the usual Zariski sheaf cohomology that were observed under base change: "natural definitions of S-schemes $X \to S$, which in classical geometry gave vector bundles X over S, did not have in general the property of being 'locally' products of a (Zariski) neighborhood and a 'typical' fiber" [27, pg. 865]. The Weil conjectures point to these later axiomatizations. Unfortunately, I cannot develop any of those axiomatizations here, except to say that they go deep into scheme theory.

What we can say with the tiny bit that we have learned? Returning to Question 18: the choice of base field k obviously affects the number of solutions we can expect to find for an arbitrary ideal in $k[x_1, ..., x_n]$. But as the Weil conjectures make clear, the choice of base field is also important because it has an important and subtle effect on the way we organize and define the systems of invariants—i.e. cohomology theories in different sheaves, of different dimensions, over different classes of points and open subsets—sitting above any algebraic variety. In turn, these systems of invariants are literally defined by the interaction between the algebra of the base field and the geometry embedded within the base topology. To be clear, *each* cohomology theory defines a different system of invariants.[32]

A last remark. Given what we have already said about Question 16, the next step is to study the different, possible topological axiomatizations. We have seen sheaf cohomology defined on top of the Zariski topology, but it is also possible to pivot the other way: to use sheaves as an indirect means of defining new topological axiomatizations, this time over categories rather than topological spaces. Witness Grothendieck's étale topology, which was inspired by the étale space of the structure sheaf. But for now, lacking more specifics, I will not speculate further.

[32] Indeed, one can see a similar interaction between 'base field' and 'mechanism of abstraction' happening in the earlier foundations of the subject: the simplification of projective geometry allowed by adding complex points of intersection allowed mathematicians to see much more clearly the underlying organization of phenomena like the intersection of conics, leading to later abstractions like the first applications of permutation and symmetry group, via Klein's program.

> “In mathematics, there are not only theorems. There are, what we call, “philosophies” or “yogas,” which remain vague. Sometimes we can guess the flavor of what should be true but cannot make a precise statement. When I want to understand a problem, I first need to have a panorama of what is around it. A philosophy creates a panorama where you can put things in place and understand that if you do something here, you can make progress somewhere else. That is how things begin to fit together.” - Deligne

> “[A] theory of induction is superfluous. It has no function in a logic of science. The best we can say of a hypothesis is that up to now it has been able to show its worth, and that it has been more successful that other hypotheses although, in principle, it can never be justified, verified, or even shown to be probable. This appraisal of the hypothesis relies solely upon deductive consequences (predictions) which may be drawn from the hypothesis: There is no need to even mention induction.” - Chervonenkis

5 Organizing principles in machine learning

Machine learning is the art and science of constructing and reconstructing mathematical models from data. It encompasses forms of statistical inference like Bayesian inference and various forms of deep learning, though not all mathematical models constructed from data take the form of probability distributions. A *learning algorithm* A is a formal method for constructing and reconstructing mathematical models from data. The mathematical models constructible by A often belong to a particular class, which we call the *concept class* C of A. For the sake of concreteness, I restrict myself to mathematical models in the sense of *concepts* $c : X \to \{-1, +1\}$ (equivalently: subsets of X) for a binary classification task on a sample space, X. (Though eventually I would like to move away from the idea of ‘tasks’ to focus solely on the behavior.) So a concept class is just some subset of 2^X.

This thesis seeks to develop a higher-methods perspective on machine learning in response to the following conjecture:

Conjecture 20 (Gromov [48], informal)**.** There exists a universal learning algorithm.

The conjecture is not about the nature of human or artificial intelligence, whatever its inspirations. To me, it says something deep about the nature of mathematical models and how they can be reconstructed from data. Posed in a different way:

Question 21. Does there exist a general-purpose, behavioral classification of learning algorithms? Thus, implicitly: does there exist a general language for comparing, constructing, and gluing concept classes?

The premise of this thesis is that the the answer to Question 21 is *yes*. That is, there should exist different systems—emphasis on *systems*—of statistical and behavioral invariants through which we can classify and equate learning algorithms according to their output over roughly "equivalent" data, in a sense yet to be made precise. This would allow us to combine and compose learning algorithms, in a way that the current, relatively ad hoc use of single invariants (mostly to bound the expected error of the learners) does not. In particular, I would like to have a more intrinsic, general-purpose definition of the concept class of A that is independent of the embedding into a sample space. For example, if A is polynomial regression, then the concept class of A should be a polynomial ring, not just the graphs of polynomials in a sample space X with a set of fixed features as basis. But as yet, I am still far from realizing this vision.

Remark 5.1. Long experience in AI tells us that general-purpose definitions are a bad idea, in the following sense: the real world is too complex, and to solve a practical problem that is embedded in the real world, any solution must also be embedded, tested, and iterated in the world. Just as there is no mathematical technique that works best in every circumstance, there is no single, static knowledge representation—mathematical or otherwise—through which one can interpret the world. The world must speak for itself.

On the other hand, much of computational learning theory tries to be *representation-independent*. For example, "good" results in computational learning theory are often *distribution-free*, i.e. they characterize the learnability of concept classes for all distributions, *completely independent of what the data actually looks like*, (though it's hardly true that distributional / parametric approaches really think about what the world looks like either, at least in any deep or structural way) through combinatorial invariants like the Vapnik-Chervonenkis (VC) dimension and the Rademacher complexity, accuracy and confidence parameters as in traditional PAC theory, and traditional complexity measures like space and time complexity. Moreover, building a learning algorithm that is independent of the representation or parameterization can be very practically important, since many learning algorithms are quite sensitive to the choice of representation. For example, witness the performance of ridge regression (which penalizes large coefficients) on data centered around 0, versus data shifted away from 0 by some constant. Every time we normalize data, we are witnessing the practical importance of representations.

The plan, in brief

The behavior or output of a learning algorithm is the concept (or, more rarely, the concept class) that it constructs on a given sample set. In the second half of this thesis, I will focus on developing the relationship between two mechanisms for describing concept classes: the VC dimension and sample compression schemes. These two are related by a more traditional problem of computational learning theory: the *sample compression conjecture*:

Conjecture 22. (Littlestone & Warmuth [65]) Any concept class C of VC-dimension d possesses a sample compression scheme of size $\mathcal{O}(d)$.

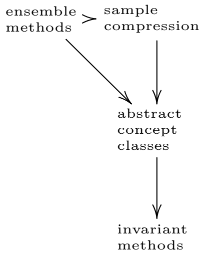
So the universal learning conjecture motivates this thesis “from the top-down”, as the search for organizing principles for learning algorithms. The sample compression conjecture motivates this thesis “from the bottom-up”, as a series of technical advances relating to a specific combinatorial invariant of concept classes (the VC dimension) and building off existing attacks on the conjecture. However, it’s not easy (for me) to see how category theory and sheaf theory can be used productively for either conjecture. My initial, naive attempts were far from fruitful. So, working with Samson, I’ve been honing the first part of my thesis: to directly study AdaBoost, an existing, popular method for combining and composing learning algorithms, using sheaf theory. AdaBoost has been analyzed using the VC dimension (though there are some suggestive defects in arguments based on the VC dimension), but it can also be interpreted as particular form of sample compression [94, Ch. 4.2].

Concretely, the thesis will unfold through three technical papers to be written up in Hilary, Trinity, and Michaelmas of 2019.

1. Sheaf cohomology for AdaBoost. See Section 5.4.

2. Cubical complexes on finite domains. See Section 5.5.

3. Invariant methods for machine learning. See Section 5.6.



Sections 5.1 through 5.3 constitute the literature review. Section 5.1 covers some essential terms and ideas from computational learning theory. Section 5.2 introduces the sample compression conjecture, along with the most recent geometric attacks on it. Section 5.3 reviews ensemble methods, especially the AdaBoost algorithm of Freund and Schapire [39] and its relationship to sample compression.

I sketch the rough plan of the thesis in Sections 5.4 through 5.6.

5.1 A very brief review of computational learning theory

Computational learning theory was developed in the 60s to analyze the convergence and efficiency properties of learning algorithms, starting with Novikoff’s bound [85] and Minsky's (in)famous attack on the venerable percetron [79], and gave rise to several practical approaches to learning: PAC theory to boosting, VC theory to SVMs. Most of the following definitions come from Mohri [80].

In computational learning theory, a *concept* c is a labeling function $c : X \to Y$ that assigns to each possible input $x \in X$ a label $y \in Y$. In the case of binary classification, $Y = \{+, -\}$, and we commonly think of concepts as the positively labeled subsets of X. Usually we will be concerned not with particular concepts but with an entire concept class C—a learning algorithm for C is one capable of differentiating concepts in C and choosing the right (or best) concept given the data. Often, we will also refer to concepts as hypotheses and concept classes as hypothesis spaces, especially if there is some "true" concept that we are trying to approximate.

Example 5.2. The class of axis-aligned rectangles on $X = \mathbb{R}^2$ is a concept class $C \subset \{+, -\}^{\mathbb{R}^2}$ given by all labelings of data such that the positive examples $c^{-1}(+)$ are contained in the interior of an axis-aligned rectangle.

In the 1980s, Valiant [115] developed the theory of *probably approximately correct (PAC) learning*, which is a framework for analyzing classification algorithms with respect only to the concept class and the cardinality of the sample set (many PAC results are *distribution-free*, meaning true for all choices of distribution). An algorithm A can *PAC-learn* a concept class $\mathcal{C}$ if it almost always $(p \geq 1 - \delta)$ picks a concept c that is approximately correct $(error(c) < \epsilon)$. Specifically,

Definition 5.3. For a given learning algorithm A, a concept class C is *PAC-learnable* by A if, given a sample set S drawn from D of size m lesser than or equal to a polynomial in $1/\epsilon, 1/\delta, n$, and size(c), we have

$$\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta$$

where $R(h_S)$ is the expected error of the hypothesis h_S (generated by A) on the distribution D, and size(c) is an extra parameter representing the cost of the computational representation of $c \in C$. ϵ is called the *accuracy* and δ is called the *confidence*. The sample size m is called the *sample complexity* of the algorithm. (A bit of notation: in machine learning, $\Pr_{S \sim D^m}(E)$ is shorthand for the probability of event E, given that S is obtained from drawing i.i.d. m examples from the distribution D.)

We say that C is merely *learnable* if $\infty > m \geq \text{poly}(1/\epsilon, 1/\delta, n, \text{size}(c))$.

Most results in PAC learning are confined to finite concept classes, while most learning problems of interest require one to distinguish between infinitely-many concepts with only a finite number of samples. One general strategy for lifting PAC-type results to the infinite case is to establish the result for a family of (finite) concept classes. There are three related ways of doing this: via the *Rademacher complexity* of C, via the *growth function* of C, and via the *VC dimension* of C.

Definition 5.4. Given a sample $S = (z_1, ..., z_m)$ and a class of real-valued functions C on the domain Z, the *empirical Rademacher complexity* of C given S is

$$\mathrm{Rad}_S(F) = \frac{1}{m}\mathbb{E}\left[\sup_{c\in C}\sum_{i=1}^{m}\sigma_i f(z_i)\right]$$

where σ_i are independent random variables drawn from the Rademacher distribution, i.e. $\Pr(\sigma_i = +1) = \Pr(\sigma_i = -1) = 0.5$.

Definition 5.5. The *growth function* $\Pi_C : \mathbb{N} \to \mathbb{N}$ for a concept class C on a domain X is defined by:

$$\Pi_C(m) = \max_{\{x_1,...,x_m\}\subset X} |\{(c(x_1), .., c(x_m)) : c \in C\}|$$

Equivalently, $\Pi_C(m)$ is the maximum number of distinct ways in which m points from X can be classified using concepts in C.

Note that the highest value that Π_C for a fixed m can achieve is 2^m, i.e. all possible binary labelings of m points.

Example 5.6. Let $X = \mathbb{R}$. Let Σ_1 be the concept class of threshold functions

$$c(x) = \begin{cases} +1 & \text{if } x \geq \nu \\ -1 & \text{otherwise} \end{cases}$$

where ν is a constant. Then $\Pi_{\Sigma_1}(m) = m + 1$.

Example 5.7. Let $X = \mathbb{R}$, and consider the concept class K of unions of intervals, i.e. $c(x) = +1$ on a finite set of intervals and -1 on the complement. Then $\Pi_K(m) = 2^m$ for all m.

The growth function is often used to bound the expected error using various variants of the following lemma.

Lemma 5.8. *For any given concept class C and any sample set S of size m, with probability* $1 - \delta$

$$R(c) \leq \frac{2\log_2(\Pi_C(2m)) + 2\log_2(2/\delta)}{m}$$

for every $c \in C$ consistent with S.

Note that variants of the above lemma exist where c may not be completely consistent with S, i.e. $\hat{R}(h) > 0$.

Definition 5.9. We say that a sample set S of size m is *shattered* by a concept class C when all 2^m possible labelings can be realized by concepts in C. That is, S is shattered by C if $\Pi_C(|S|) = 2^m$.

Definition 5.10. The *Vapnik-Chernovenkis (VC) dimension* of a concept class $C \subset \{0,1\}^X$ is

$$\mathrm{VC}(C) = \max \{m : \Pi_C(m) = 2^m\}$$

In other words, the VC dimension of C is the cardinality of the largest set of points that C can shatter.

Lemma 5.8 has an analogous formulation in terms of the VC dimension.

Lemma 5.11. *For any given concept class C and any sample set S of size m, with probability $1-\delta$*

$$R(c) \leq \frac{2d \log_2(2em/d) + 2\log_2(2/\delta)}{m}$$

for every $c \in C$ consistent with S.

While VC dimension is defined only for binary classification, note that there are similar notions for multiclass problems, i.e. $|Y| > 2$.

By a basic result called Sauer's lemma, a concept class $\mathcal{C}$ is PAC-learnable if and only if it has finite VC dimension. We will come back to Sauer's lemma in the next section.

term	definition / notation
concept or hypothesis	$h : X \to Y, \quad Y = \{-1, +1\}$
loss function	$L(a,b)$, penalty for $a \neq b$
expected error or test error	$R(h) = \mathbb{E}_{(x,y)\sim D}[L(h(x), y)]$
empirical error or training error	$\hat{R}_S(h) = \frac{1}{m}\sum_{i=1}^m L(h(x_i), y_i)$
generalization error	$R(h) - \hat{R}(h)$
zero-one loss	$L_{0-1}(a,b) = \begin{cases} 0 & a = b \\ 1 & a \neq b \end{cases}$
square loss	$L_{sq}(a,b) = (1-ab)^2$
hinge loss	$L_{hin}(a) = \max(0, 1-ab)$
logistic loss	$L_{\log}(a,b) = \log(1+e^{-ab})$
exponential loss	$L_{\exp}(a,b) = e^{-ab}$
sample complexity	m, number of examples needed to PAC-learn a given concept

Table 5.1: A glossary of basic mechanisms for describing the behavior of a learning algorithm A with hypothesis h on a given (labeled) data set S.

5.2 A very brief review of sample compression

For a given concept class $C \subset 2^X$, a (unlabeled) *sample compression scheme of size k* is a pair of maps (f,g) where f is a compression function mapping finite sample sets S from X to compressed subsets of examples of size $\leq k$, called *compression sets*, and g is a reconstruction function mapping each such compression set to a concept consistent with S.

$$f : (X \times Y)^m \to X^{\leq k}$$
$$g : X^{\leq k} \to C$$

Note that in certain variants, g need not return a concept in C, as long as the concept is consistent with the original hypothesis.

The founding paper is Littlestone & Warmuth's [65]. In it, they defined sample compression schemes, described the connection to PAC learning, and articulated the sample compression conjecture:

Conjecture 23. (Littlestone & Warmuth [65]) Any concept class C of VC dimension d admits sample compression schemes of size $\mathcal{O}(d)$.

Translation:

1. *VC dimension* d: in the worst case, we can learn with poly(d) examples
2. *schemes of size* d: in the best case, we can learn with d examples

Of course, having a sample compression scheme of size d for a concept class almost immediately gives us a class of VC dimension d [65]. But can we use the VC dimension, usually used to define "worst case" upper bounds on a learning algorithm, to construct a strict bound on the number of examples we have to use in the best case, with the "best" examples?[33]

Example 5.12 (Axis-aligned rectangles). The easiest example of an (unlabeled) sample compression scheme is that for axis-aligned rectangles in $\mathbb{R}^2$. Given a sample set consistent with some rectangle, the compression function f compresses the sample to the topmost, bottommost, rightmost, and leftmost (positive) examples. The reconstruction function g returns the tightest-fit axis-aligned rectangle around these examples. So we have a sample compression scheme of size 4=VC(C) for axis-aligned rectangles. See Section 5.2.

A sample compression scheme of size k is a proof of the proposition that at most k examples are needed to describe a given concept $c \in C$—that is, as a kind of minimum-length description. The point is that sample compression is about *samples*. But in general, we can explicitly define the compression function f and the reconstruction function g to include bit strings $\in M$ with arbitrary information:

$$f : (X \times Y)^m \to X^{\leq k} \times M$$
$$g : X^{\leq k} \times M \to C.$$

For example, there exists a trivial compression scheme whose compression function outputs the empty compression set and a bit string that encodes the

[33] The best-worst comparison isn't exactly right, since we do not assume, in a typical VC analysis, that we receive the "worst" examples. For a strict contrast, we might look at adversarial learning.

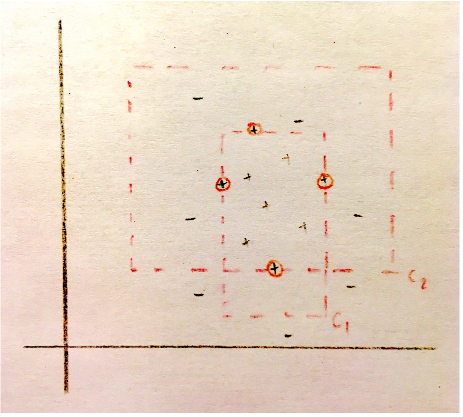


Figure 5.1: Axis-aligned rectangles as concepts. Concepts c_1, c_2 are given by the dotted boxes, and $d(c_1 + c_2)$ is the boundary of $c_1 + c_2$ given by the four circled points.

minimum-length description of the given concept, and a reconstruction function that interprets the bit string back into the original concept.

Let S be the sample set, and let σ be a message string. If $g(S, \sigma)$ ignores σ, then we have recovered the original definition of sample compression schemes. If $g(S, \sigma)$ ignores S, then we have returned to standard statistical learning theory [**?**].

A specific variant of generalized compression schemes are *labeled compression schemes*, which output not only the compression set but also the labels in $\{0, 1\}$:

$$f_{\text{labeled}} : (X \times Y)^m \to (X \times Y)^{\leq k}$$

$$g_{\text{labeled}} : (X \times Y)^{\leq k} \to C$$

Historically, labeled compression schemes were considered before (unlabeled) compression schemes.

Example 5.13 (Halving). Suppose X is a finite domain. Then the Halving algorithm of Angluin and Littlestone [64] suggests a simple labeled compression scheme for samples from X called the *one-pass compression scheme* [36]. Recall that the Halving algorithm keeps track of all concepts consistent with the past examples in a "version space" and, given a new example, predicts the label based on a majority vote of all concepts in the version space. It then "halves" the concepts in memory by deleting those that were wrong on the example.

Suppose that we modify the Halving algorithm so that it culls the concepts in its version space only on mistakes; then for every example that it predicts incorrectly, we save that example to a (labeled) compression set $A^{\pm}$ (whose elements are of the form (x_i, y_i) where $x_i \in X$ and $y_i \in \{-1, 1\}$). We can then use the Halving algorithm to reconstruct the original concept. Since the Halving algorithm has a mistake bound of $\log |C|$ for any concept class C on

X, this immediately shows that we have a sample compression scheme of size at most $\log |C|$ for any finite concept class C.

Before we get too hopeful, note that the compression scheme here cannot help us show the sample compression conjecture since it depends only on the cardinality of C (as opposed to the VC dimension). Clearly a concept class of constant VC dimension could still be of arbitrary size, and thus have a sample compression scheme of arbitrary size.

Example 5.14 (SVM). Support vector machines (SVMs) lead to a sample compression scheme for halfspaces $\{x \in \mathbb{R}^n : w \cdot x \geq b\}$, since it suffices to keep just the set of (essential) support vectors, along with their labels, as we can infer the maximum-margin hyperplane in dimension n from a set of $n+1$ essential support vectors. As expected, $n+1$ is also the VC dimension of the concept class of halfspaces of dimension n.

The combinatorial picture

Usually we think of VC dimension as a combinatorial invariant of the concept class (e.g. as the size of the maximum shattering set of a concept class), and this interpretation goes along with a basic result called the *Sauer-Shelah lemma*, often referred to as Sauer's lemma:

Lemma 5.15 (Sauer-Shelah)**.** *Let $\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i}$, then if $C \subset 2^X$ is a concept class on X with finite VC dimension d, we have*

$$\Pi_C(|S|) \leq \Phi_d(|S|) \text{ for all samples } S \subset X \text{ of size } m,$$

where the growth function Π_C, discussed in Section 5.1, gives the number of different ways to classify a finite subset S using concepts in C.

Sauer's lemma, which was proved independently in statistical learning, combinatorics, and model theory, directly implies that a concept class is PAC-learnable if and only if it has finite VC dimension (e.g., see [13]).

To color in this combinatorial sketch of VC dimension, we define maximum classes:

Definition 5.16. A concept class $C \subset 2^X$ is *maximum* if $\Pi_C(|S|) = \Phi_d(|S|)$ for all finite subsets S of X. That is, every finite subset of X satisfies Sauer's lemma with equality.

Example 5.17. From [61]: a finite maximum class of VC dimension 2 with 11 concepts, along with their behavior on a sample set of 4 (maximally distinct) examples $x_1, x_2, x_3, x_4 \in \mathbb{R}^n$.

Example 5.18. The class of positive halfspaces (halfspaces that contain the point $(\infty, 0, ..., 0)$) and negative halfspaces (halfspaces that contain the point $(-\infty, 0, ..., 0)$) are almost maximum, in that the restriction to (any shattering class of) any set of points in general position will produce a maximum class.

Example 5.19. The class of axis-aligned rectangles is *not* maximum.

	x_1	x_2	x_3	x_4
c_1	0	0	0	0
c_2	0	0	1	0
c_3	0	0	1	1
c_4	0	1	0	0
c_5	0	1	0	1
c_6	0	1	1	0
c_7	0	1	1	1
c_8	1	0	0	0
c_9	1	0	1	0
c_{10}	1	0	1	1
c_{11}	1	1	0	0

Maximum classes are, in a way, the most well-structured classes of those with finite VC dimension. Typically, we use the VC dimension to say something about the concept class; e.g. the growth function is often exponential in $|S|$, so having finite VC dimension says a lot about the "niceness" of C. To say that the growth function and Φ_d are equal allows us to use maximum classes to say something about the VC dimension; in fact many facts about VC dimension can be translated into facts about maximum classes, which will be helpful because *maximum classes are always compressible*. Several current attacks on the compression conjecture exploit this feature of maximum classes; the hope is to find a way of embedding arbitrary classes of VC dimension d into maximum classes of VC dimension $\mathcal{O}(d)$.

The compressibility of maximum classes has to do with the fact that maximum classes have a nice, global structure that allows them to be contracted "along the sample" to a trivial concept class: a class having only one concept. The use of the word "contracted" is no accident; beautifully, maximum classes over a finite domain $X = \{0,1\}^n$ have a very convenient geometric structure: we can view them as *cubical complexes* (so, just subsets of an n-cube), and the existence of a (certain sort of) k-compression scheme as synonymous with k-contractibility [93]. So there is at least a proof-of-concept of a productive use of topological methods in this area.

For a more detailed introduction to sample compression (with more proofs), consider [36] or Floyd's thesis [35]. For more details on the geometric approaches to maximum classes, we encourage the reader to look at [61] for an introduction and [93] for the latest research. Moran [82] recently demonstrated that, for any concept class (whether or not over a finite domain), there exists a sample compression scheme of at least order exponential in the VC dimension.

While this thesis does not really aim at the compression conjecture, it is a helpful and concrete motivation—a stone against which we can sharpen our intuitions and questions about learning. Many practical machine learning algorithms are based on finding compression sets, e.g. SVMs and kernel machines. Less well-known are the applications to AdaBoost, discussed below.

5.3 A very brief review of AdaBoost

PAC learning constructs *strong* learners, in the sense that we can (almost) always construct a hypothesis with arbitrarily small empirical error, given a reasonable number of examples. But strong learners may be difficult to construct depending on how complicated the concept class is, and they may have other undesirable properties, e.g. high computational complexity. On the other hand, *weak* learners, which need to do only better than guessing, can often be constructed quite easily.

Definition 5.20. For a given learning algorithm A, a concept class C is *weakly PAC-learnable* by A if, given any sample set $S \sim D^m$, it almost always picks a hypothesis $h \in C$ that is better than random guessing on X. That is,

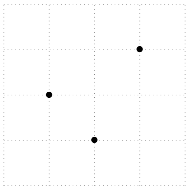
$$\Pr_{S \sim D^m}[R(h, S) \leq \frac{1}{2} - \epsilon] \geq 1 - \delta, \quad \epsilon, \delta > 0.$$

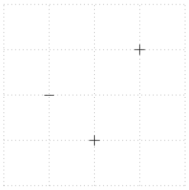
As usual, ϵ and δ are the accuracy and confidence parameters, respectively.

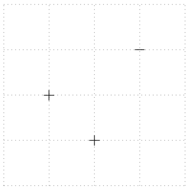
For a given concept class C, we call such a learning algorithm A a *weak learner* on C, and the output of A a *weak classifier.*

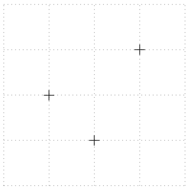
Boosting is a class of techniques for combining or constructing strong learners from many weak learners, though in practice, it is only used to combine the hypotheses of many instances of one weak learning algorithm. Following convention, we refer to those methods involving multiple hypotheses (called the *base classifiers*) of one base learner as *ensemble methods*, while those that involve different learners as *multiple classifier systems.* At least in the first part of the thesis, we will be dealing solely with ensemble methods.

Example 5.21. Given $\{h_1, h_2, h_3\}$ as hypotheses on X, where

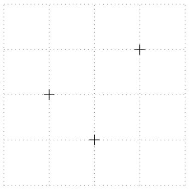
$X =$ 

$h_1(X) =$ 

$h_2(X) =$ 

$h_3(X) =$ 

we have an obvious ensemble method given by

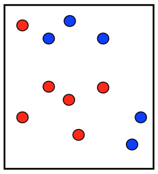
$\texttt{majority\_vote}(h_1, h_2, h_3)(X) =$ 

Aside from the weak learning condition, ensemble methods are defined by the *diversity* of their base classifiers. Roughly, a set of base classifiers is *diverse* if their errors are uncorrelated; diverse classifiers, when wrong, tend to get different examples wrong. The importance of diversity is obvious; it doesn't make much sense to combine many copies of the same classifier. Many ensemble methods optimize for diversity, e.g. bagging. Unfortunately, as of 2014 [121], there is still no agreed-upon definition of diversity or how to measure it [26, 60, 109]. We will come back to this problem in Section 5.4.

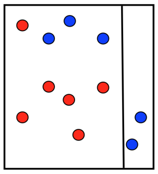
Among the variety of boosting methods, perhaps the most famous and robust algorithm is AdaBoost, introduced by Yoav Freund and Robert Schapire in [39]. Uniquely among boosting methods, AdaBoost maintains a distribution over not only the set of base classifiers, but also a distribution over the sample set S. The distribution over the base classifiers weighs the relative importance or trustworthiness of each base classifier. The distribution over the sample set S tells us how to count errors on S.

Rather than go through the pseudocode directly (see Algorithm 1), let's walk through a quick example using axis-aligned hyperplanes as our base classifiers.

Example 5.22. The following is borrowed from [80]. Start with a sample set S, as below.

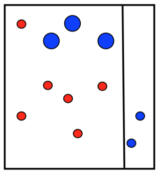


AdaBoost proceeds in rounds (called boosting rounds). In round $t = 1$, every example in S has the same weight. We ask the weak learner for a classifier, and it returns the following hyperplane:



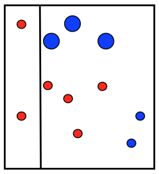
$t = 1$

First, we calculate a error coefficient, α_1, that characterizes how badly h_1 did on the sample, given the distribution at $t = 1$. We then increase or decrease the weights on S accordingly: more if h_1 got the label wrong, and less if it got it right.

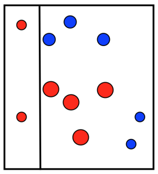


$t = 1$

At $t = 2$, we repeat the procedure with the new weights.

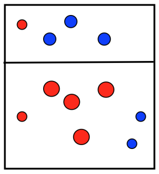


$t = 2$

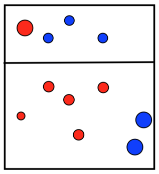


$t = 2$

And so on, for $t = 3$.

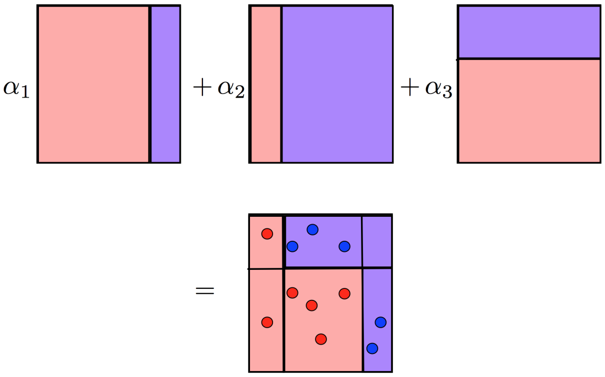


$t = 3$



$t = 3$

We then iterate for an arbitrary number of rounds of boosting, T. For $T = 3$, the output of AdaBoost is of the form



Remark 5.23. Axis-aligned hyperplanes are typically associated with decision stumps, a.k.a. boosting stumps, which are the most commonly used base learner in AdaBoost. With decision stumps, the total computational complexity of AdaBoost is

$$\mathcal{O}(mN \log m + mNT) = \mathcal{O}(m \log m)$$

where m is the size of the sample, N is the dimension of the space, and T is the number of rounds of boosting [80]. Of course there are many different options for base learner, e.g. [63] used kernel SVMs by conditioning the RBF kernel on a parameter σ, the "Gaussian width", but decisions stumps tend to have the best across-the-board performance.

Most treatments of AdaBoost, especially theoretical ones, do not assume any particular weak learner. Instead, they often pick the concept class.

It remains to review some basic results on the empirical error and generalization error of AdaBoost, before treating the connection between AdaBoost and sample compression.

Algorithm 1 AdaBoost

```
procedure AdaBoost($S, T, A$)                          ▷ Assuming labels in $Y = \{-1, +1\}$
    $m \leftarrow |S|$                                  ▷ $S$ is the sample set
    for $i = 1$ to $m$ do $D_1(i) \leftarrow 1/m$       ▷ Initialize the weights on $S$
    for $t = 1$ to $T$ do                               ▷ $T$ rounds of boosting
        $h_t \leftarrow A(S, \vec{D}_t)$                ▷ The weak learner $A$
        $\epsilon_t \leftarrow \sum_{i=1}^{m} D_t(i) y_i h_t(x_i)$    ▷ Error term, weighted on $\vec{D}_t$
        $\alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon}\ t$    ▷ $\alpha_t : (0, 1) \to \mathbb{R}$
        $Z_t \leftarrow 2(\epsilon_t(1 - \epsilon_t))^{1/2}$    ▷ Normalization term
        for $i = 1$ to $m$ do
            $D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} \exp(-\alpha_t y_i h_t(x_i))$    ▷ Update new weights
    $g \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
    return $h \leftarrow \mathrm{sign}(g)$
```

Empirical error

Theorem 5.24 (Freund and Schapire [39]). *For h returned by AdaBoost and any S of size m,*

$$\hat{R}_S(h) \leq \exp(-2 \sum_{t=1}^{T} (\frac{1}{2} - \epsilon_t)^2$$

i.e. the empirical error on S decreases as an exponential function of T.

Further, if for all $t \in [1, T]$ we have a constant $\gamma \leq (1/2 - \epsilon_t)$, then

$$\hat{R}_S(h) \leq \exp(-2\gamma^2 T)$$

i.e. γ (called the edge*) gives a uniform convergence result on $\hat{R}(h)$.*

Proof. I assume the following small lemma:

$$D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m \prod_{s=1}^{t} Z_s}$$

Then

$$\begin{aligned}
\hat{R}(h) &= \frac{1}{m}\sum_{i=1}^{m} \#(y_i \neq h(x_i)) && && (1)\\
&\leq \frac{1}{m}\sum_{i=1}^{m} e^{-y_i g(x_i)} && \text{note the replacement of h with g} && (2)\\
&= \frac{1}{m}\sum_{i=1}^{m} \left[m \prod_{t=1}^{T} Z_t D_{T+1}(i) \right] && \text{by the lemma above} && (3)\\
&= \prod_{t=1}^{T} Z_t && && (4)\\
&= \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} && \text{by definition of } Z_t && (5)\\
&= \prod_{t=1}^{T} \sqrt{1-4(\frac{1}{2}-\epsilon_t)^2} && && (6)\\
&\leq \prod_{t=1}^{T} \exp[-2(\frac{1}{2}-\epsilon_2)^2] && \text{since } \sqrt{1-x} \leq 1-x \leq e^{-x} && (7)\\
&= \exp\,[-2\sum_{t=1}^{T}(\frac{1}{2}-\epsilon_t)^2] && && (8)
\end{aligned}$$

□

What's going here? First off, it's clear that the empirical error goes down very quickly. In experiments, the empirical error usually disappears within several rounds of boosting. But perhaps more importantly, the theorem gives us a good amount of insight into the structure of AdaBoost, and especially into the definitions of α_t (Line 7 of Algorithm 1), Z_t (Line 8), and the update rule for D_{t+1} (Line 10). In effect, Z_t and the update rule are derived from the choice of α_t, which itself was defined in order to minimize a particular loss function—interpreted in the theorem, Eq. (7), as an upper bound on the zero-one loss $1-x$—called the *exponential loss function*,

$$\text{loss}_{\text{exp}}(h) = \sum_{x_i \in S} e^{-y_i h(x_i)}.$$

There are other loss functions, e.g. the logistic loss (see Section 5.3)

$$\text{loss}_{\text{log}}(h) = \sum_{x_i \in S} \log(1 + e^{-y_i f(x_i)})$$

which correspond to other versions of boosting, e.g. LogitBoost [40].

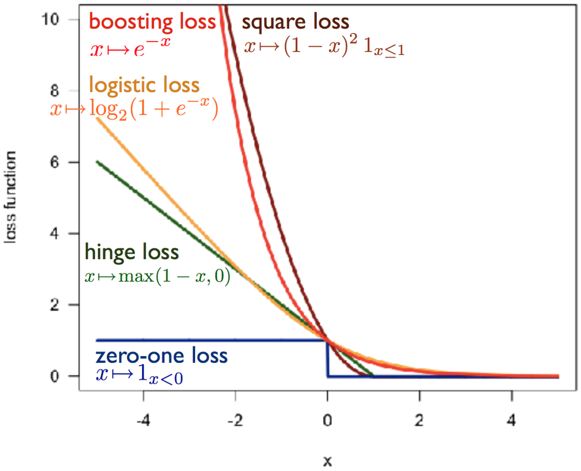


Figure 5.2: Different loss functions from [80], all of which can be interpreted as upper bounds over the zero-one loss. AdaBoost is equivalent to coordinate descent on the boosting loss, also known as the exponential loss.

Generalization bounds via VC dimension

Generalization error, as discussed in Section 5.1, is the difference between the expected error $R(h)$ of hypothesis h on all of X and the empirical error $\hat{R}_S(h)$ on a finite sample $S \subset X$. Even though we have shown that AdaBoost minimizes the empirical error, this is not the same as showing that it will minimize the expected error on all of X, which is the goal of boosting and of PAC learning more generally.

One common way of upper bounding the generalization error is by measuring the complexity of the concept class. The VC dimension is a way of doing this for infinite concept classes. For finite concept classes we do not need to resort to the VC dimension, but the derivation is still quite instructive.

Recall that the hypotheses output by AdaBoost are of the form $h = \text{sign}(g) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t)$. Let C_T be the concept class representing all functions of that form. (Note: since $\alpha_t \in \mathbb{R}$, C_T is infinite, whether or not C is finite.)

The following is Theorem 4.3 in [94].

Theorem 5.25. *Suppose AdaBoost is run for T rounds on $m \geq T$ random examples, using base classifiers from a finite concept class C. Then, with probability at least $1-\delta$ (over the choice of the random sample), the combined classifier h satisfies*

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{32[T\log_2(\frac{em|C|}{T}) + \log_2(8/\delta)]}{m}} \sim \hat{R}(h) + \mathcal{O}(\frac{T\log m|C|/T}{m})$$

Furthermore, with probability at least $1-\delta$, if h is consistent with the training *set (so that $\hat{R}(h) = 0$), then*

$$R(h) \le \frac{2T\log_2(\frac{2em|C|}{T}) + 2\log_2(2/\delta)}{m} \sim \mathcal{O}(T\frac{\log(m|C|/T)}{m}.$$

The theorem tells us that the bound on the expected error is in terms of the empirical error $\hat{R}$, the number examples m, and two terms that stand in for the complexity of C_T: the number of concepts in $|C|$ and the number of boosting rounds T.

For infinite concept classes, we have the following theorem (Theorem 4.6 in [94]):

Theorem 5.26. *Suppose AdaBoost is run for T rounds on $m \ge T$ random examples, using base classifiers from a concept class C of VC dimension $d \ge 1$. Then, with probability at least $1-\delta$ (over the choice of the random sample), the combined classifier h satisfies*

$$R(h) \le \hat{R}(h) + \sqrt{\frac{32[T\log_2(\frac{em}{T}) + d\log_2(\frac{em}{d})\log_2(8/\delta)]}{m}}$$

Furthermore, with probability at least $1-\delta$, if h is consistent with the training set (so that $\hat{R}(h) = 0$), then

$$R(h) \le \frac{2T\log_2(\frac{2em}{T}) + d\log_2(\frac{2em}{d}) + 2\log_2(2/\delta)}{m}.$$

Note the obvious analogy with the previous theorem.

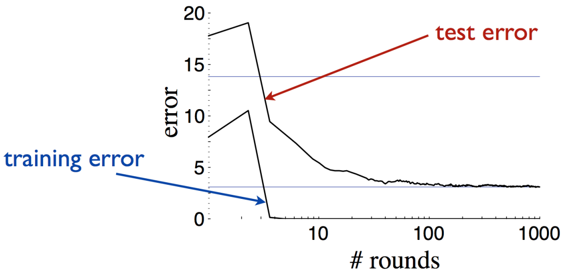
Remark 5.27. If the base classifiers are drawn from a concept class of VC dimension d, then the VC dimension of the combined classifiers in C_T output by AdaBoost (after T rounds of boosting) is at most

$$\mathrm{VC}(C_T) = 2(d+1)(T+1)\log_2((T+1)e).$$

In particular, this implies that

$$R(h) \sim \mathcal{O}(dT\log_2 T).$$

In other words, the number of expected errors increases as T increases; VC analysis tells us that AdaBoost will tend to overfit for large T. However, in actual experiments, AdaBoost does *not* tend to overfit. In fact, the generalization performance tends to improve even for $T \gg 0$.



So the VC dimension offers only a weak sort of bound.

Generalization bounds via margin analysis

The idea: margin-based methods quantify the "confidence" of the combined hypothesis, which continues to increase even when the empirical error goes to 0.

Definition 5.28. Let $\hat{C}_T$ be the set of functions of the form $g = \sum_{t=1}^{T} \alpha_t h_t$. The *margin* of a point (x, y) with respect to $g \in \hat{C}_T$ is

$$\begin{aligned} \text{margin}_g : X \times Y &\to [-1, 1] \\ (x, y) &\mapsto yg(x). \end{aligned}$$

Definition 5.29. The ℓ_1-*margin* $\rho(x)$ of a point (x, y) with respect to $g \in \hat{C}_T$ is

$$\rho(x) = \frac{yg(x)}{\sum_{t=T}^{m} |\alpha_t|} = y\frac{\vec{\alpha} \cdot \vec{h}(x)}{||\alpha||_1}.$$

We let $\rho_g = \min_{i \in [1,m]} y_i \frac{\vec{\alpha} \cdot \vec{h}(x)}{||\alpha||_1}$.

What's going on here? In effect, $\rho(x)$ is measuring the $|| \cdot ||_\infty$ distance between $\hat{h}(x)$ and the hyperplane $\vec{\alpha} \cdot \vec{x} = 0$.

The key result of margin-based methods: generalization error is *independent* of T! In particular, with probability $1 - \delta$,

$$R(g) \leq \hat{R}_\rho(g) + \frac{2}{\rho} \text{Rad}_m(C) + \sqrt{\frac{\log_2 1/\delta}{2m}}$$

where $\text{Rad}_m(C)$ is the Rademacher complexity of C. Note that this result is true for any $\frac{g}{||\alpha||_1}$ which lies in the convex hull C_T, for *any* concept class C.

But it's not that AdaBoost achieves the maximum margin or works as a maximum-margin optimizer. In fact, empirically, it seems to beat algorithms that strictly optimize the ℓ_1-margin. But margin-based methods help us understand boosting as a way of adaptively fitting and adjusting the loss/margin criterion. We will address this fact in Section 5.4.

Generalization bounds via sample compression

In Section 5.2, we focused on how to define and construct sample compression schemes for different concept classes, abstracting away from learning algorithms completely. But of course, many algorithm do take the form of sample compression schemes, and these tend to benefit from certain guarantees. Just as the VC dimension can be thought of as a description of how much information (in the form of a shattering set of size m) is needed to distinguish between two concepts in a concept class, a sample compression scheme of size k tells us that only k points are needed to distinguish between any two concepts.

In particular, we have the following theorem:

Theorem 5.30 ([94, Ch. 4.2])**.** *Suppose a learning algorithm based on a labeled compression scheme of size k is provided with a random training set S of size m. Then with probability at least $1-\delta$, any hypothesis h consistent with S produced by this algorithm satisfies*

$$R(h) \leq \frac{k \log_m + \log(1/\delta)}{m-k}.$$

Freund and Schapire [94, Ch. 4.2] showed that AdaBoost is based on a certain kind of labeled compression scheme.

The idea: we start with a severely constrained version of AdaBoost with the following changes (in order of severity)

1. deterministic weak learner: the weak learner does not employ randomization, so it is a deterministic mapping from a sequence of examples to a hypothesis

2. resampling: instead of treating the current distribution D_t as a weight on S, on each round of boosting, the weak learner is trained on an *unweighted sample* of size $m_0 < m$ drawn with replacement from D_t over S

3. unweighted vote: the combined classifier is a simple *unweighted* majority vote, i.e. $\alpha_1 = \alpha_2 = ... = 1$

This constrained version is clearly a sample compression scheme of size $k = Tm_0$, since each combined classifier could be uniquely represented by a compression scheme of size Tm_0. The problem is with the last assumption: clearly, AdaBoost does not output an unweighted majority vote. But how can we "compress" a hypothesis involving real-valued weights $\alpha_1, ..., \alpha_T$ down to a finite set of examples?

In response, Freund and Schapire introduce *hybrid compression schemes*: compression schemes where the reconstruction function returns not a single hypothesis but an entire concept class. How it works: given a sample set S of size m, AdaBoost first "compresses" S (i.e. resampling according to D_t) down to a sequence $((x_1, y_1), ..., (x_k, y_{m_0}))$ in S. Actually, it does this T times, sequentially and each time with resampling from $D_t, t \in [1, T]$. This larger sequence $((x_1, y_1), ..., (x_k, y_{Tm_0}))$ of size Tm_0 is the true compression set. AdaBoost then "reconstructs" the hypothesis *class*—not hypothesis!—of all weighted majority vote classifiers over the *fixed* set of base classifiers $h_1, ..., h_T$:

$$\sigma_T = \{h(x) = \sum_{t=1}^{T} \alpha_t h_t : \alpha_1, ..., \alpha_T \in \mathbb{R}\}.$$

It's not clear from [94, Ch. 4.2] if the story simple ends there; there is no treatment of how AdaBoost picks an actual hypothesis from this class. There may be some sort of generalized compression scheme, where the D_t are treated as message strings, then used to reconstruct the α_i, that captures the full behavior of AdaBoost, Given that AdaBoost is a compression scheme, we can use (a slightly modified version of) Theorem 5.30 to demonstrate the following bound:

Theorem 5.31 ([94, Ch. 4.2])**.** *Suppose AdaBoost is run for T rounds on $m \geq (m_0 + 1)T$ random examples. Assume that we have a deterministic weak learner and resampling as above. Then, with probability at least $1 - \delta$ (over the choice of the random sample), the combined classifier h, assuming it is consistent with S, satisfies*

$$R(h) \leq \frac{2T \log_2(2e(m - Tm_0)/T) + 2Tm_0 \log_2 m + 2\log(2/\delta)}{m - Tm_0} \sim \mathcal{O}(\frac{T \log_2 m}{m - T}).$$

The problem with this bound? Aside from the fact that T is still a "bad" number in the bound (the error goes up as T increases), the bound is slightly boring. It looks a lot like bounds we've already seen from the VC dimension; we've simply replaced one complexity measure of hypotheses, denominated in terms of the concept class, with another, denominated in terms of examples.

Bias and variance

There exists a tension between the accuracy of the weak learner and the diversity of the ensemble. That is, since more accurate learners make less mistakes (on the training data), ensembles of accurate learners will tend to agree on more examples. Additionally, even when they make mistakes, accurate learners tend to make similar mistakes (reflecting intrinsic noise or outliers). This tradeoff is reflected in a more well-known problem in machine learning: the *bias-variance tradeoff*. In particular, there is a classic decomposition of the expected error called the bias-variance decomposition:

$$R(h) = \text{Bias}(h) + \text{Var}(h) + \epsilon \tag{9}$$

where ϵ is some measure of the irreducible error (e.g. from noise) in the data, bias is the expected loss between the "average" hypothesis of the learner and the true hypothesis, and variance is a measure of the error arising from the sensitivity of the learner to small fluctuations in the sample set, i.e. overfitting.[34] Fact: a more accurate learner has less bias. Intuitively at least, we expect a more diverse ensemble to have less variance and thus less risk of overfitting.

[34] Compare Eq. (9) with the typical form of the generalization bounds we have seen:

$$R(h) \leq \hat{R}(h) + \text{complexity}(C) + \text{confidence}(\delta). \tag{10}$$

If we assume that the combined hypothesis h of AdaBoost is consistent with the training data, we can ignore the empirical error $\hat{R}(h)$ in Eq. (10). By the weak learning criterion and Theorem 4.28 in [94], we can ignore the bias term in Eq. (9); with sufficient rounds of boosting and large enough m, we can always approximate the true concept on X. So a typical bound on the variance of AdaBoost should look something like the following:

$$\text{Var}(h) \leq \text{complexity}(C) + \text{confidence}(\delta) - \epsilon \tag{11}$$

Interpretation: the amount that h will overfit is bounded by the complexity of the concept class C it is drawn from, the confidence of the hypothesis, with some allowance for intrinsic noise.

Not unexpectedly (given the difficulty of defining diversity), there is a lot controversy over the right definitions of bias and variance. In a way that is still unclear to me, obtaining a proper diversity measure is likely at least somewhat related to obtaining a proper definition of bias and variance.

As an example, consider one such definition from [28]:

Definition 5.32. For a fixed learning algorithm and fixed x, let $y_m \in Y$ be the *main prediction* given by

$$y_m = \text{argmin}_{y'} \mathbb{E}_{S \sim D^m} \big[L(h_S(x), y') \big].$$

Note that S is not fixed above, so h_S ranges over all hypotheses generated by the learner for all possible S.

Definition 5.33. The *main hypothesis* is the hypothesis h_m which outputs the main prediction for all $x \in X$.

Note that h_m is not necessarily in the concept class of the given learning algorithm.

Definition 5.34. For a given learner, the *bias* of a hypothesis on X is

$$\text{Bias}(h) = \mathbb{E}_{(x,y) \sim D} \big[L(h_m(x), y) \big].$$

Definition 5.35. For a given learner, the *variance* of a hypothesis on X is

$$\text{Var}(h) = \mathbb{E}_{(x,y) \sim D, S \sim D^m} \big[L(h_m(x), h_S(x)) \big]$$

Note the dependence on the particular loss function in both the variance and the bias.

5.4 Sheaf cohomology for AdaBoost

Everything in this section, should be marked ‘speculative’, in the sense of likely to change or to be wrong.

So far, we have reviewed boosting, an ensemble approach to learning that combines many instances of a weak learner into a strong learner. The hypotheses of the weak learner are then combined in some way, e.g. in a linear combination, in order to output a hypothesis, and the relative weight of each hypothesis is adjusted over many rounds of boosting, based on their errors on training examples. AdaBoost, in particular, “compresses” samples at each iteration (whether through re-weighting or resampling) so as to focus attention on those examples which are harder to classify; namely, on those which previous rounds of the weak learner have misclassified. From the perspective of compression, AdaBoost learns the corners or boundaries of the data set through successive rounds of boosting, which it then uses to guide the construction of a strong learner with low generalization error.

The heft behind most theoretical guarantees for AdaBoost comes from the weak learner assumption; indeed, we have almost never invoked the update step in Line 10 of Algorithm 1 (the one exception: in the first guarantee on the empirical error) except through T, the number of rounds of boosting. In the margin-based analysis, even T itself was done away with. This makes sense in a way, since similar guarantees apply to other forms of boosting. In a game-theoretic analysis, [38] uses von Neumann's minmax theorem to prove that the weak learning condition is equivalent to saying that the data is linearly separable[35] (with ℓ_1 margin), and a further refinement [94, Ch. 4.3] allows us to prove an even stronger statement: a concept class is weakly PAC-learnable if and only if it is (strongly) PAC-learnable. So what we thought was a relatively weak assumption turns out to be quite a strong assumption: in principle, it is just as hard to build weak learners as it is to build strong learners!

But ensemble methods like AdaBoost are also characterized by the *diversity* of their base classifiers. Clearly, AdaBoost tends to increase diversity over rounds of boosting: according to [57], "AdaBoost's choice of the new distribution can be seen as an approximate solution to the following problem: Find a new distribution that is closest to the old distribution subject to the constraint that the new distribution is orthogonal to the vector of mistakes of the current weak hypothesis." Unfortunately, as of 2014 [121], there is still no agreed-upon definition of diversity or how to measure it in an ensemble [26, 60, 109]. Further, ensemble methods that strictly try to maximize diversity have relatively poor performance; [109, pg. 256] explains this as "the minimum margin of an ensemble is not monotonically increasing with the diversity."

Example 5.36. Three examples, collected from [109] and [60]. Let h_j, h_k be two base classifiers in an ensemble, S a sample set of size m, T the number of base classifiers, and let $n(a,b) = n_{j,k}(a,b)$ be the number of training examples in S on which h_j outputs a and h_k outputs b.

1. The *disagreement measure* between h_j, h_j is

$$\text{dis}_{j,k} = \frac{n(1,-1)+n(-1,1)}{n(1,1)+n(-1,1)+n(1,-1)+n(-1,-1)}.$$

   This is one of the most straightforward measures. The diversity of the entire ensemble is then the average,

$$\text{dis} = \frac{2}{T(T-1)} \sum_{t=1}^{T} \sum_{k=t+1}^{T} \text{dis}_{j,k}.$$

2. The *entropy measure* E is one among many based on the idea that, for a particular $x \in X$, the highest diversity is when half the base classifiers predict -1 and the other predict $+1$. Let $l(x_i)$ be the number of base classifiers that correctly classify x_i. Then

$$E = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{T - \text{ceiling}(T)} \min\{l(x_i), L - l(x_i)\}$$

[35] Note that "linearly separable" here means that X can be classified, with no error at all, with a linear combination of weak learners, i.e. a hyperplane defined by $\vec{\alpha} \cdot \vec{h}$, *not* that X can be separated by a typical hyperplane in $\mathbb{R}^n$.

3. Let $l_i = T \sum_{h_t : h_t(x_i) \neq y_i} \alpha_t$, i.e. the weighted vote of all incorrect classifiers on x_i. The *Kohavi-Wolpert variance* of an ensemble is then defined as

$$KW = \frac{1}{mT^2} \sum_{i=1}^{m} l_i(T - l_i).$$

   This measure is adapted from the bias-variance decomposition of the error of a classifier (see below).

Empirically, learning algorithms constructed to maximize these and other diversity measures all have similar performance [60]. They also all underperform AdaBoost, implying that AdaBoost is not just trying to maximize the diversity of its ensemble. The state-of-the-art explanation for the generalization performance of AdaBoost is that it tends to maximize the minimum margin, thus increasing the confidence of a hypothesis even after the empirical error goes to 0 [94]. To be clear, this is a high bar; much of the generalization performance of the algorithm *can* be explained by margin-based methods. But still, we know that AdaBoost is *not* maximizing the margin, and that it even outperforms boosting algorithms that strictly optimize the ℓ_1 margin. So there is something left to explain: "perhaps a finer analysis based on a different notion than that of margin could shed light on the properties of the algorithm" [80].

Question 24. Can we use diversity measures to provide a finer analysis of AdaBoost than margin-based methods?

My proposal:

(1) characterize diversity measures by their *contextuality*, and

(2) use cohomological arguments to reconstruct existing generalization bounds.

(1) builds on a suggestion of David Balduzzi in [10], who proposed an analysis of distributed systems (e.g. neural systems, or cellular automata) using what he called structure presheaves, which are similar to the measurement presheaves $D_R \circ \mathcal{E}$ in [2]. (I've written up a brief discussion of this approach in Appendix A.) Indeed, in an unpublished research statement [9], Balduzzi proposes a research program to study ensemble methods as distributed systems using sheaf cohomology—with the first example to be an analysis of AdaBoost! Unfortunately, I'm not aware of any further work in this vein. So I would like to start by realizing Balduzzi's proposal.

5.4.1 Background

Let S be a finite subset of a topological space X. We call S the *sample set*. Let $Y = \{-1, +1\}$, called the labels, and let $H = \{h_1, ..., h_T\} \subset Y^X$ be a finite set of binary classifiers on X. Give H the discrete topology.

Definition 5.37. The *sample presheaf* $\hat{S}$ of S is the following functor:

$$\begin{aligned}\hat{S} : \mathrm{Open}(H)^{\mathrm{op}} &\to \mathbf{Set}\\ U &\mapsto \hat{S}(U) := \{\hat{x} : U \to Y :: h \mapsto h(x), x \in S\}\\ U \subset V &\mapsto \mathrm{res}_{V,U} : \hat{S}(V) \to \hat{S}(U) :: \hat{x} \mapsto \hat{x}_{|U}\end{aligned}$$

Each $\hat{x} \in \hat{S}(U)$ captures the labels $h_t(x_i)$ assigned by all classifiers in U to the example $x_i \in S$.

Lemma 5.38. *$\hat{S}$ does not satisfy the sheaf gluing axiom, but does satisfy the uniqueness axiom.*

Proof. It suffices to consider a simple counterexample with $S = \{x, y\}$, $H = \{h_1, h_2, h_3\}$ as below, and a cover $\mathfrak{U} = \{U = \{h_1, h_2\}, V = \{h_2, h_3\}\}$.

	x	y
h_1	-	+
h_2	+	+
h_3	+	-

Then $\hat{S}(H) = \{\hat{x}, \hat{y}\}$, $\hat{S}(U) = \{\hat{x}|_U, \hat{y}|_U\}$, and $\hat{S}(V) = \{\hat{x}|_V, \hat{y}|_V\}$. The sections $\hat{x}|_U \in U$ and $\hat{y}|_V \in V$ form a compatible family over $\mathfrak{U}$, since they agree on $U \cap V = \{h_2\}$, but the glued function given by $h_1 \mapsto -1, h_2 \mapsto +1, h_3 \mapsto -1$ does not correspond to any element in $\hat{S}(H)$, since there is no $s \in S$ taking those values in the table. So the gluing axiom fails.[36]

On the other hand, suppose there exists such a global section $\hat{z} \in \hat{S}(H)$ for some compatible family over $\mathfrak{U} = \{U_i\}_{i \in I}$. Further, suppose there exists another such global section $\hat{z}' \in \hat{S}(H)$ which restricts to the same local sections. Then $\hat{z}(h) = \hat{z}'(h)$ for all $h \in U_i$. Since the U_i cover H, $\hat{z}(h) = \hat{z}'(h)$ for all $h \in H$, therefore $\hat{z} = \hat{z}'$ in $\hat{S}(H)$, showing uniqueness. □

In particular, each $\hat{x} \in \hat{S}(U)$ is defined by an equivalence class $[x] \subset S$, where the equivalence relation is $x \sim_U x'$ iff $h(x) = h(x')$ for all $h \in U$, and any global section over a compatible family $\{\hat{x}_i\}_{i=1,\ldots,k}$, if it exists, can be represented by some $x \in [x_1] \cap \ldots \cap [x_k]$. There is no global section precisely when $[x_1] \cap \ldots \cap [x_k]$ is empty.

From a machine learning perspective, however, the fact that $\hat{x} \in \hat{S}(U)$ is defined by an equivalence class of points is a problem. That is, $\hat{S}$ is throwing away some important information about S: namely the multiplicity of examples satisfying any classification scheme U. Below, we compose $\hat{S}$ with a distribution monad in order to capture this multiplicity (but secretly for another reason:

[36] One should view the failure of the gluing axiom not as saying something about S (e.g. it's not large enough) but as saying something about H. A failed gluing indicates that there is behavior that an ensemble of hypotheses could achieve (or "simulate"?) which has not been yet observed by the hypotheses, run individually.

AdaBoost maintains a distribution over S).[37] Later, we will try out a simpler functor, F_R that directly counts the number of examples without normalizing things into a distribution; this turns out to work better for purposes of cohomology. (Though I am not convinced this is the best way; it may be simpler to pass to multi-sets. There may also be a more topological solution that retains more information from X.)

Let R be a commutative semiring (typically $\mathbb{R}_{\geq 0}$). Let

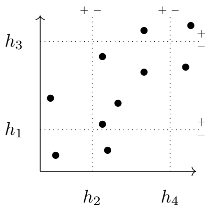
$$\begin{aligned} D_R &: \mathbf{Set} \to \mathbf{Set} \\ &A \mapsto D_R(A) = \text{the set of } R\text{-distributions over } A \\ A \xrightarrow{f} B &\mapsto D_R(A) \overset{D_R(f)}{\to} D_R(B) :: d \mapsto [b \mapsto \sum_{f(a)=b} d(a)] \end{aligned}$$

Even when $\hat{S}$ is a sheaf, $D_R \circ \hat{S}$ may not be a sheaf: the gluing axiom does not hold (ref. contextual models). However, the fact that D_R is only a **Set**-valued functor will be problem later, when we want to invoke cohomology.

Example 5.39. Let $H = \{h_1, h_2, h_3, h_4\}$ and fix a cover

$$\mathfrak{U} = \{\{h_1, h_2\}, \{h_2, h_3\}, \{h_3, h_4\}, \{h_4, h_1\}\}.$$

Let S be a set of 10 points in X as in the graph below. Then S generates the following (noncontextual) empirical model for $\mathfrak{U}$ taking values in $D_R \circ \hat{S}$.



$\mapsto$

	$(+1,+1)$	$(+1,-1)$	$(-1,+1)$	$(-1,-1)$
$h_1\ h_2$	1/10	7/10	1/10	1/10
$h_2\ h_3$	0	2/10	2/10	6/10
$h_3\ h_4$	1/10	7/10	1/10	1/10
$h_4\ h_1$	6/10	2/10	2/10	0

Figure 5.3: We are given a 10 examples $x \in S$ in X, as above. The dotted lines represent hypotheses (in this case, axis-aligned hyperplanes) that classify part of X as positive and part as negative. The topology of H then generates the table to the right: each row of the table is a distribution, reflecting the locations and proportions of the sample set S with respect to a set of hypotheses, i.e. an element of the cover $\mathfrak{U}$.

[37] One way of interpreting an empirical model / global distribution $\in D_R \circ \hat{S}$ of is to take it as an approximation of the original, "true" distribution on X (or $X \times Y$), from which S was drawn. It is an approximation in two senses: both in terms of accuracy/confidence (we may have gotten unlucky with S) and in terms of precision (the regions marked by H cannot capture the true shape of the distribution). Alternately (and I would argue more naturally), we can take the empirical model / global distribution as a distribution on S itself, i.e. as an assignment of (normalized) weights to each example in S "up to what H can distinguish". Indeed, this is what AdaBoost does: AdaBoost maintains a distribution D_t on S at each boosting round t.

The distributions specified on each row of Example 5.39 then glue to form the obvious global distribution on $\hat{S}(H)$, which has exactly 7 elements, corresponding to the 7 regions in the graph with non-zero support.[38]

5.4.2 An analogy

Imagine a set of classrooms where students are taking practice exams. The questions on the exam are all yes/no questions. In some classrooms there is one teacher, t, and one student, h. In others, there are two students, h_1 and h_2, and one teacher. In some rooms, there are only students. The students can form study groups (of pairs, threes, etc.) in which they compare notes. However, none of the students know the right answers, so the best they can do is track which questions they agree or disagree on. Some questions are easy for the students (or so difficult and/or poorly-written that they shouldn't have been placed on the exam): every student agrees on them. Some questions are hard for the students: 50% of the students choose yes and 50% choose no. The teacher can also "compare notes", but she does this by putting the answers up on the blackboard so that everyone can compare themselves against the answers.

And the students aren't monkeys: they're not guessing randomly. They have some sort of prior on the subject of the exam, though they may have different (incorrect) understandings of the subject.

In this analogy, the practice exam is the sample set S, the "final" exam is the input space X, poorly-written questions represent outliers in the data, the teacher represents the true classification $t = t_S$, the students are the hypotheses $h \in H$, classrooms are open covers $\mathfrak{U}$ of H, and study groups are elements of the cover $U_i \in \mathfrak{U}$. Sections $\hat{x} \in \hat{S}(U)$ correspond to sections (and sometimes individual questions) on the practice exam, and compatible families of sections $\{\hat{x}_i\} \in \prod \hat{S}(U_i)$ represent artfully-designed questions—not necessarily on the practice exam S!—that can tease out the differences between different students. Compatible families of distributions $\{d_i\} \in D_R \circ \hat{S}(U)$ represent sections of the practice exam weighted by some criterion, e.g. the total number of questions in that section or the number of questions that students get wrong.

The fact that students aren't monkeys corresponds to the fact that hypotheses in machine learning aren't (usually) just maps $[n] \to \{-1, +1\}$. Just as a student has a prior on the subject of an exam, so a hypothesis is defined on the entire input space X of a sample set S; further, these hypothesis are usually characterized by some structure built on the topology of X: e.g. axis-aligned rectangles in $\mathbb{R}^2$, or hyperplane separators in $\mathbb{R}^n$ as in Example 5.39.

Below, I will discuss a conjectural distance function $\hat{L}$ between compatible families of sections. $\hat{L}$ represents "learning".

[38] The shaded cells represent regions of X where no $s \in S$ exists. That is: the local distribution for $\{h_2, h_3\}$ *is not even defined* on the region of X where $h_2(x) = +1$ and $h_3(x) = +1$, since there is no such section $\hat{s} \in S(\{h_i, h_j\})$, since there is no $s \in S$ in that region. Note also that some regions have 0 area, e.g the region $\{x \in X : h_1(x) = -1, h_3(x) = +1\} = \emptyset$. Again, these 0-area regions are not even assigned a probability (not merely probability 0), since we are composing with $\hat{S}$ (or even $\hat{X}$).

5.4.3 Cohomology

Following [3], compatible families of distributions live in the 0-th cohomology group $H^0(\mathfrak{U}, D_R \circ \hat{S}(U)$. So we would like to construct the nerve $\mathcal{N}(\mathfrak{U})$ of $\mathfrak{U}$ and the (Cech) cohomology of the presheaf $D_R \circ \hat{S}$ with respect to $\mathfrak{U} = \{U_i\}_{i \in I}$. The problem is that D_R is not a presheaf of abelian groups.

To make the cohomology well-defined, i.e. to define addition and subtraction on chains, we pass to an abelian presheaf $F_{\mathbb{Z}} \circ \hat{S}$, where $F_{\mathbb{Z}}$ is the functor

$$\begin{aligned} &F_{\mathbb{Z}} : \mathbf{Set} \to \mathbf{Set} \\ &\qquad X \mapsto F_{\mathbb{Z}}(X) = \{\phi : X \to \mathbb{Z} : \phi \text{ has finite support}\} \\ &X \xrightarrow{f} Y \mapsto F_{\mathbb{Z}}(X) \overset{F_{\mathbb{Z}}(f)}{\to} F_{\mathbb{Z}}(Y) :: \phi \mapsto [y \mapsto \sum_{f(x)=y} \phi(x)] \end{aligned}$$

We denote by $|[-]|$ the section in $F_{\mathbb{Z}} \circ \hat{S}(U)$ which assigns to each $\hat{x}$ the cardinality of the equivalence class $[x]$.

The finite support condition is not necessary here; in practice S is always finite.

Example 5.40. Let $X = S = \{x, y, z\}$, $Y = \{-1, +1\}$, $H = Y^X$, and $\mathfrak{U}$ be the set of all singletons in H. Then $\mathcal{N}(\mathfrak{U})$ has $|H| = 8$ vertices, representing each singleton in $\mathfrak{U}$, and no edges. In dimension 0, the corresponding cochain complex has

$$C^0(\mathfrak{U}, \hat{S}) = \hat{S}(\{h_1\}) \times ... \times \hat{S}(\{h_8\}).$$

Since each h_i is a Boolean function on $X = S$, $C^0(\mathfrak{U}, \hat{S})$ has exactly 14 elements (16 - 2 maps that are identically +1 or -1 on X). We think of each element as a vertex labeled with a map $h_i \mapsto \pm 1$.

Example 5.41. In the above example, let

$$\mathfrak{U} = \{U_i\}_{i=1,\dots,8} = \left\{ \{h_1, h_2\}, \{h_2, h_3\}, ..., \{h_7, h_8\}, \{h_8, h_1\} \right\}.$$

As before, $\mathcal{N}(\mathfrak{U})$ has 8 vertices labeled with the elements of the cover, and 8 edges. Then $C^0(\mathfrak{U}, \hat{S}) = \hat{S}(\{h_1, h_2\}) \times ... \times \hat{S}(\{h_8, h_1\})$ has between 17 and 20 elements depending on how the h_i are ordered; each such element represents a *pair* of maps $h_i \mapsto \pm 1$, $h_j \mapsto \pm 1$. $C^1(\mathfrak{U}, \hat{S}) = \hat{S}(\{h_1\}) \times ... \times \hat{S}(\{h_8\})$ has exactly 14 elements, labeled with maps $h_i \mapsto to \pm 1$.

Pick a family of functions $\vec{\phi} = (\phi_1, ..., \phi_8) \in C^0(\mathfrak{U}, F_{\mathbb{Z}} \circ \hat{S})$; each component function ϕ_i can be thought of as a formal $\mathbb{Z}$-linear sum of elements in $\hat{S}(U_i)$. The coboundary operator $\delta^0 : C^0(\mathfrak{U}, F_{\mathbb{Z}} \circ \hat{S}) \to C^1(\mathfrak{U}, F_{\mathbb{Z}} \circ \hat{S})$ acts on $\vec{\phi}$ and $\sigma = U_1 \cap U_2 \in \mathcal{N}_1(\mathfrak{U})$ by

$$\begin{aligned} \delta^0(\vec{\phi})(\sigma) &= \sum_{k=1}^{2} (-1)^k \text{res}_k(\vec{\phi})(\hat{S}\partial_k \sigma) \\ &= \text{res}_1(\vec{\phi})(\hat{S}U_1) - \text{res}_2(\vec{\phi})(\hat{S}U_2) \end{aligned}$$

where the restriction homomorphism $\text{res}_1 : F_{\mathbb{Z}} \circ \hat{S}(U_2) \to F_{\mathbb{Z}} \circ \hat{S}(U_1 \cap U_2)$ can be thought of as distributing the sum $\phi_i \in F_{\mathbb{Z}} \circ S(U_2)$, and similarly with res_2. In other words, $\text{res}_1(\vec{\phi})(\hat{S}U_1)$ is the number of examples counted by $\vec{\phi}$ in each region of X, represented by $\hat{x} \in \hat{S}(U_1 \cap U_2)$ after "gluing" some regions together. Just as in [3], the 0-dimensional cohomology classes $H^0(\mathfrak{U}, F_{\mathbb{Z}} \circ \hat{S})$ represents compatible families of sections over $\hat{S}(U_i)$ for each element U_i of the cover.

One thinks of the sheaf $F_{\mathbb{Z}} \circ \hat{S}$ as a tool for quantifying the extent to which different hypothesis sets disagree on regions of X (which are approximated by S), weighted by how important that region is according to $\vec{\phi}$. In machine learning terms, the sheaf allows us to quantify, using test data from S, how independent each hypothesis is from any other.

The cohomology measures to what extent these families of formal sums disagree with each other. The problem is that the empirical models / distributions assumed in traditional machine learning are not contextual, so the higher cohomology, while present, is not interesting.[39]

5.4.4 Conjectures

A typical error bound (e.g. in VC analysis, or through margin-based methods, or through compression) on AdaBoost looks like the following:

$$\text{err}(h) \leq \hat{\text{err}}_S(h) + \mathcal{O}\Bigg(\text{complexity}(\mathbf{H}_A, T) + \text{confidence}(\delta)\Bigg)$$

where $\text{err}(h)$ is the expected error (with respect to a fixed loss function) of the combined hypothesis $h = \text{sign}(\sum_{h_i \in H} \alpha_i h_i) = \text{sign}(\vec{h} \cdot \vec{\alpha})$ output by AdaBoost, $\hat{\text{err}}_S$ is the empirical error with respect to S, $\mathbf{H}_A$ is the hypothesis class output by the weak learner A, $T = |H|$ is number of base classifiers generated by AdaBoost, and $\delta > 0$ is a fixed parameter in $\mathbb{R}$.

The goal of this paper is to reproduce (and hopefully improve on) traditional bounds on the expected error of AdaBoost and other ensemble methods. Traditional approaches to constructing the error bound focus on the complexity term, but these approaches run into the bias-variance tradeoff discussed elsewhere in the proposal: simpler concept classes can underfit the training sample, thus increasing the empirical error $\hat{\text{err}}_S$, but more complex classes can also overfit the data, raising the expected error err on all of X. The strategy here is orthogonal: enrich the notions of expected error and empirical error so that they naturally carry a notion of efficiency or "excess information" [10] between the base classifiers in H. In other words, we want to redefine err and $\hat{\text{err}}_S$ from functions on single hypotheses to functions on *systems* of hypotheses, while preserving the general format of traditional error bounds:

$$\text{err}(H, \vec{\beta}) \leq \hat{\text{err}}_S(H, \vec{\beta}) + \mathcal{O}\Bigg(\text{complexity}(\mathbf{H}_A, |H|) + \text{confidence}(\delta)\Bigg)$$

[39] There may, however, be a link between contextual models and models of *active learning*, in which learners can choose to explore different parts of the sample space.

where $\vec{\beta} = (\beta_1, ..., \beta_k) \in \prod F_{\mathbb{Z}} \circ \hat{S}(U_i)$ is a compatible family of sections with respect to a given cover of H. (Note: while $\vec{\alpha}$ can be thought of as a weighted sum of $h \in H$, $\vec{\beta}$ is a family of weighted sums on $s \in S$ "with respect to H".) In order to be useful, $\mathrm{err}(H, \vec{\beta})$ should upper bound the expected error $\mathrm{err}(h)$ of the combined hypothesis $h = \mathrm{sign}(\sum_i \alpha_i h_i)$.

For a given sample S, we introduce "error" by simply adjoining to H (and to $\mathfrak{U}$) a partial function $t : X \to Y :: x \in S \mapsto \mathrm{oracle}(x)$, representing the true classification of $x \in S$.[40] Let $H_t = H \cup \{t\}$ and $\mathfrak{U}_t = \mathfrak{U} \cup \{t\}$. Given the definition of $\hat{S}$, it does not matter that t is a partial function on X, so long as it is defined on S.

Definition 5.42. The *error surface* of a fixed loss function[41] L is the graph in $\mathbf{H}_A \times \mathbb{R}$ of the empirical error $\hat{\mathrm{err}}_S(h)$, where $\hat{\mathrm{err}}_S(h) = \sum_{s \in S} L(t(s), h(s))$.

Many learning algorithms A (e.g. decision trees, or neural networks, or SVMs), identify $\mathbf{H}_A$ with a parameter space, usually some subset of $\mathbb{R}^n$. n may be *very* large, depending on the algorithm and the dimension of the input space X. In what follows, we will assume that $\mathbf{H}_A \simeq \mathbb{R}^n$.

Definition 5.43. For a set of base hypotheses H and a cover $\mathfrak{U}$, the *error nerve* $E(\mathfrak{U})$ of $\mathfrak{U}$ with respect to t is the cone of the nerve of $\mathfrak{U}$, letting t be the vertex of the cone.

Simplices corresponding to facets of the cone do not have a natural interpretation, as in the rest of the nerve, as intersections between their vertices: for example, U_i and $\{t\}$ may have empty intersection. We need to define a slightly modified version of the (Cech) cohomology of a presheaf with respect to a cover $\mathfrak{U}$.

Definition 5.44. The *weak error cohomology* $H^n_{\mathrm{err}}(\mathfrak{U}_t, F_{\mathbb{Z}} \circ \hat{S})$ is the Cech cohomology of the following cochain complex: on p-simplices σ in $\mathcal{N}(\mathfrak{U})$, p-cochains are defined as usual: as $\mathbb{Z}$-formal sums of sections in $F_{\mathbb{Z}} \circ \hat{S}(|\sigma|)$. On simplices introduced by the cone construction, a p-cochain valued in $F_{\mathbb{Z}} \circ \hat{S}$ is a $\mathbb{Z}$-formal sum of just those sections which are correctly classified (according to t) by *at least half*[42] of the hypotheses in $|\sigma| = U_{i_0} \cap ... \cap U_{i_{p-1}}$.

[40] Sometimes this true classification is introduced with the distribution, i.e. S is considered as a finite subset of $X \times Y$ drawn according to a distribution on $X \times Y$. My impression is that this is a piece of technical and notational convenience rather than a thought-out position. Conceptually, the mechanism for drawing an X "from nature" and then labeling it with some $y \in Y$ are distinct.

[41] In classification, a *loss function* is a function of the form $L : Y \times Y \to \mathbb{R}$ which represents the cost of a bad prediction. There is no precise definition of a loss function, though there is a standard list of common loss functions. For example, the fundamental loss function is the *zero-one loss* $L_{0-1}(y, y') = \delta_y(y')$, where δ_y is the Dirac delta function on y. (We say fundamental because other loss functions are sometimes thought of as surrogates for L_{0-1}.) However, L_{0-1} is non-convex and non-smooth, so it is often better to approximate it using other loss functions which are continuous, convex, and more computationally tractable.

[42] This is inspired by AdaBoost's weak learning assumption, but I still need to work out an example.

The error nerve is meant to serve as a simplicial / combinatorial approximation of the error surface and thus, by extension, of the empirical error $\hat{\text{err}}_S(h)$. So the immediate technical objective is to define a method of embedding the error nerve—or a similar construction—into $\mathbf{H}_A \times \mathbb{R}$ in order to verify that the nerve does, indeed, approximate the surface.

Conjecture 25. Fix a cover $\mathfrak{U}$ over H. Then every loss function L induces a metric

$$\hat{L} : \mathfrak{U} \times \mathfrak{U} \to \mathbb{R}$$

between elements of the cover $\mathfrak{U}$ such that if $U_i \cap U_j$ is non-empty, then

$$\hat{L}(U_i, U_i \cap U_j) + \hat{L}(U_j, U_i \cap U_j) \leq \hat{L}(U_i, U_j).$$

In particular, $\hat{L}$ should commute with "taking compatible families" in the following sense: for any compatible family of sections $\vec{\phi}$ over $\mathfrak{U}$ in $F_{\mathbb{Z}} \circ \hat{S}$, there exists a unique family of hypotheses h_{ϕ_i} over the U_i, such that

$$L(h_{\phi_i}, h_{\phi_i|_{U_i \cap U_j}}) + L(h_{\phi_j}, h_{\phi_j|_{U_i \cap U_j}}) \leq L(h_{\phi_i}, h_{\phi_j}).$$

Intuitively, the loss between two singletons $\{h_i\}, \{h_j\}$ should be

$$\hat{L}(\{h_i\}, \{h_j\}) = \sum_{s \in S} L(h_1(s), h_j(s)),$$

while the loss between two arbitrary elements of the cover U_i, U_j should be thought of as the total loss incurred over S between two *combined* hypotheses

$$h_i = \sum_{h_k \in U_i} \alpha_k h_k$$

$$h_j = \sum_{h_k \in U_j} \beta_k h.$$

The metric $\hat{L}$ will help us construct the requisite embedding into $\mathbf{H}_A \times \mathbb{R}$.

Conjecture 26. Let n be the dimension of $\mathbf{H}_A$ as a real vector space and fix a metric $\hat{L}$ on $\mathfrak{U}$ as above. Then $\hat{L}$ induces an embedding $\hat{L}_t$ of the error nerve $E(\mathfrak{U})$ into $\mathbb{R}^k$, where $k \leq n+1$ and k is some function of the homological dimension of $E(\mathfrak{U})$, taking values in $F_{\mathbb{Z}} \circ \hat{S}$.

One imagines stretching out the cone in the error nerve based on the $\hat{L}$-distance between each vertex of $\mathcal{N}(\mathfrak{U})$ and the vertex representing the true hypothesis t, which "pulls" the rest of the error nerve toward it.

Other ideas:

1. Conjecture: U is "maximally diverse" relative to H when it is maximal with respect to the VC dimension; adding one more hypothesis to U would increase the VC dimension. What is the minimum size of H so that H is able to distinguish all points in S? This is a marker of efficiency (but also of diversity).

2. Conjecture: weight adaptation can be captured by morphism of sheaves.

3. Turning a presheaf that fails uniqueness into a sheaf amounts to compression; we throw away the extra global sections (though, in actuality, throwing away global sections may not mean throwing away examples but restricting to sample sets of 'maximal' complexity). So compression *should* correspond, in the right setup, to sheafification. We'll discuss this further in Section 5.5.

4. What about "focusing attention" on a given regions of the input space X, as in active learning?

5. Typically, we call a mechanism that quantifies the difference between a presheaf and a sheaf a *cohomology theory*. Where did this statement come from; is it accurate??

6. I wonder if you can also think of certain models as complicated embeddings themselves, e.g. whether a generic word embedding to Euclidean space or even a fancier one like a Poincare embedding to hyperbolic space (in order to capture not just similarity but also hierarchical relationships, as in `https://arxiv.org/pdf/1705.08039.pdf`: "embeddings of symbolic data such that their distance in the embedding space reflects their semantic similarity... In addition to the similarity of objects, we intend to also reflect [a presumptive latent hierarchy between symbols] in the embedding space."). If so, then can you "clamp" that embedding with respect to the particular embedding into the error surface, so that as one changes, the other does as well?

> For these reasons and motivated by the discussion in Section 2, we embed symbolic data into hyperbolic space H. In contrast to Euclidean space R, there exist multiple, equivalent models of H such as the Beltrami-Klein model, the hyperboloid model, and the Poincar half-plane model. In the following, we will base our approach on the Poincar ball model, as it is well-suited for gradient-based optimization.2 In particular, let $B^d = \{x \in \mathbb{R}^d : ||x|| < 1\}$ be the open d-dimensional unit ball, where $||\cdot||$ denotes the Euclidean norm. The Poincaré ball model of hyperbolic space corresponds then to the Riemannian manifold (B^d, g_x), i.e., the open unit ball equipped with the Riemannian metric tensor
>
> $$g_x = (\frac{2}{1 - ||x||^2})^2 g^E$$
>
> where $x \in B^d$ and g^E denotes the Euclidean metric tensor. Furthermore, the distance between points $u, v \in B^d$ is given as
>
> $$d(u, v) = \text{arcosh}(1 + 2\frac{||u - v||^2}{(1 - ||u||^2)(1 - ||v||^2)})$$

7. What about combinatorial explosion in the solution space H; we need to reduce the search space in order to get reasonable algorithms by putting in a bunch of different priors. So, can the way we think about H a bit differently using the formalism above; can it give us better ways of reasoning about / organizing all the priors we need? Current solutions build *communities of agents* to find these solutions: deep reinforcement learning looks at this question, e.g. Gauthier and Mordatch 2017, to create an end-to-end policy architecture for language learning: compositional strategies in a structured environment for compositional goals.

5.5 Sample compression schemes via cubical complexes

Originally I had hoped to build a direct homological interpretation of sample compression on top of the characterization of maximum classes as cubical complexes [93]. I wanted to find a way to glue together different concept classes in a way that lets me extend useful invariants (such as VC dimension) across gluings, modeled on the kind of operations one would expect to see in differential geometry (gluing of local charts) or algebraic geometry (gluing of affine varieties). Explicitly, the idea was to associate a topological space (ideally a complex of some sort) directly to a concept class, in such a way that the topology of the space detects the VC dimension of the concept class. Then one could add concepts to the class or glue together different concept classes, and study the topology for clues to the VC dimension of the merged object. Similarly, one should be able to decompose concept classes.

In particular, I hope to build some examples using not only maximum classes, but objects called *maximal classes*: classes for which one cannot add a single additional concept without increasing the VC dimension. Maximal classes are very important in sample compression because, unlike maximum classes, we can embed all other concept classes within a maximal class of similar VC dimension.

My initial efforts failed in spectacular fashion, so I developed the strategy involving AdaBoost, above. This section is just a placeholder until I have more results in the previous section.

5.6 Invariant methods for machine learning

The original goal of all this was to develop means for combining and composing different learning algorithms. But what this really means, in order to be interesting, is to combine and compose them *over multiple domains*. So far, we have talked about adding base classifiers to create more complicated, more expressive models of a single domain, but after all, this is not quite the same thing as creating a complex model of the world. A parallel, more applied part of my research, e.g. in [103, 107, 108], concerns information integration over complex systems. Typically, this means integrating information and integrating models *by hand* across different domains X_1 and X_2. Category theory helps, but even then you cannot get out of doing most things by hand.

On the other hand, can we *learn* the right models from data? In practice, this is impossible with current techniques: there just isn't enough data. $X_1 \times X_2$ is much more complicated than X_1 and X_2 alone. For a similar reason, it's rarely feasible to just add base classifiers across domains as one does in boosting, at least without giving up all guarantees on the expected error.[43]

After finishing the research involved in the previous two papers, I would like to begin the final portion of my thesis by asking: so what *can* we say about the error in such composite, complex systems?

[43] Nonetheless, the human brain does create complex models of the world, and it does seem to add or glue the output of many separate learners into a coherent model.

A Sheaf theory for distributed systems

I am still writing up some whiteboard notes; in the meantime I've merely appended some of the definitions leading up to structure presheaves in [10]. More to follow!

Let **Stoch** be the category with objects vector spaces of real-valued functions and morphisms stochastic maps a.k.a. conditional probability distributions.[51] Fix a functor $V : \mathbf{Set} \to \mathbf{Stoch}$, so VX is a vector space of functions $\{\phi : X \to \mathbb{R}\}$ equipped with Dirac basis $\{\delta_x\}_{x \in X}$ and $V(f) : VX \to VY$ is a stochastic map commuting with V. Interpretation: $V(f)$ is a matrix of conditional probabilities $p(y|x)$.

The following definition is doing most of the key work:

Definition A.2. Given a surjective stochastic map $m : VX \to VY$, it has a *stochastic dual* $m^{\#} = m^* \circ \text{renorm} : VY \to VX$ satisfying

$$\begin{tikzcd}
(VY)^* \arrow[r, "m^*"] & (VX)^* \arrow[d, "\omega_X"] \\
(VY)\ast \arrow[u, "\text{renorm}"] \arrow[r, "\omega_Y"] & \mathbb{R}
\end{tikzcd}$$

where ω_X is the terminal map to $\mathbb{R}$ in **Stoch** (so $\omega_X^{\#}$ is the uniform distribution on X) renorm is the map which makes the columns of renorm $\circ\, m^{\#}$ sum to 1, i.e. so $m^{\#}$ is an honest stochastic map.

Note the major simplifying assumption! Interpretation: given a conditional probability distribution $m \sim p(y|x)$, the stochastic dual $m^{\#}$ computes the posterior $p(x|y)$ *assuming the uniform distribution on Y*.

Definition A.3. A *distributed dynamical system D* consists of a directed graph, an input alphabet S_l and an output alphabet A_l associated to every vertex l of D, and a state transition matrix $m_l : VS_l \to VA_l$ associated to every vertex, called a mechanism.

We can define two categories based on any such D:

[51] I think the idea is that this version of **Stoch** is a little easier to work with compared to Lawvere's original version:

Definition A.1. The category **Stoch** of stochastic processes is defined by the following data:

1. objects are measurable spaces (A, Σ_A) of sets A with a σ-algebra Σ_A
2. morphisms $P : (A, \Sigma_A) \to (B, \Sigma_B)$ are stochastic kernels, i.e. functions $P : A \times \Sigma_B \to [0, 1]$ that assign to (a, σ_B) the probability of σ_B given a, denoted $P(\sigma_B|a)$
3. composition $Q \circ P : A \times \Sigma_C \to [0, 1]$ of $P : (A, \Sigma_A) \to (B, \Sigma_B)$ and $Q : (B, \Sigma_B) \to (C, \Sigma_C)$ is defined by

$$(Q \circ P)(\sigma_C|a) = \int_{b \in B} Q(\sigma_C|b) dP_a,$$

i.e. marginalization over B

Definition A.4. The category of subsystems Sys_D on D is a Boolean lattice with objects sets of ordered pairs of vertices in D, $C \in 2^{V_D \times V_D}$, with arrows given by inclusion.

The category of measuring devices Meas_D on D has objects $\mathrm{Hom}_{\mathbf{Stoch}}(VA^C, VS^C)$ for $C \in 2^{V_D \times V_D}$, with arrows given by maps of the form

$$r_{21} : \mathrm{Hom}_{\mathbf{Stoch}}(VA^{C_2}, VS^{C_2}) \to \mathrm{Hom}_{\mathbf{Stoch}}(VA^{C_1}, VS^{C_1})$$

Definition A.5. The *structure presheaf* of a distributed dynamical system D is a functor

$$\begin{aligned} F_D : {\mathrm{Sys}_D}^{\mathrm{op}} &\to \mathrm{Meas}_D \\ C &\mapsto \mathrm{Hom}(VA^C, VS^C) \\ i_{12} : C_1 \to C_2 &\mapsto r_{21} : \mathrm{Hom}_{\mathbf{Stoch}}(VA^{C_2}, VS^{C_2}) \to \mathrm{Hom}_{\mathbf{Stoch}}(VA^{C_1}, VS^{C_1}) \end{aligned}$$

from a category of subsystems of D to a category of measuring devices whose objects are hom-spaces of stochastic maps.

Lemma A.6. *The structure presheaf satisfies the gluing axiom but not the uniqueness axiom of sheaves.*

Descent in the structure presheaf F is not unique; there are many distributions that satisfy the constraint. This is because the restriction operator is not really function restriction: it's marginalization.

B A very brief review of probabilistic programming

Other names for probabilistic programming: probabilistic databases (from Suciu et al.), and probabilistic logic programming (Poole and Sato). And a few related subjects under the rubric of "automating machine learning":

1. automatic statistician
2. Bayesian nonparametrics
3. Bayesian deep learning
4. Bayesian optimization
5. computational resource allocation
6. large-scale inference

Much of this review is drawn from https://web.cs.ucla.edu/~guyvdb/talks/UAI14.pdf.

The grand challenge in this field is "lifted inference". For example, we want to compute the number of infected people in the population, from knowing the number of sick people and the probability of contact between people in the population and the sick people. Creating such an algorithm is analogous to proving a resolution theorem in traditional logic programming.

A probabilistic program considers all the possible worlds of its program: what saves it from having to enumerate all possible worlds is that only finitely many possible worlds—a kind of compression?—are sufficient to answer any given ground query.

B.1 Learning in probabilistic programming

Learning in probabilistic programming is not very different from how it works in standard graphical models. It is just about sampling.

$$\mathrm{Pr}(\mathrm{fact}) = \frac{\#(\mathrm{fact} = \mathrm{true})}{\mathrm{all}}.$$

C A very brief review of homotopy type theory

The two key insights leading to the development of homotopy type theory (HoTT) were

1. in 2005, Awodey and Warren's interpretation of Martin-Löf type theory (MLTT) as a model category (see Section 3.4) where a type X was a kind of space—in particular, an ∞-groupoid—and the identity type of X was the *path object* of X,

2. in 2009, Voevodsky's observation that one particular model of MLTT (the category **sSet**) satisfied an important additional axiom called *univalence*.

Let $=_{\mathcal{U}}$ be the identity relation on types in a universe of types $\mathcal{U}$ and let $\simeq$ be equivalence of types. Then the *univalence axiom* states that

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B)$$

i.e. the identity relation is equivalent to the equivalence relation. Recall that the identity type of X is interpreted as the path object of X, so that $A =_{\mathcal{U}} B$ signifies a path from A to B in the universe $\mathcal{U}$. Univalence states that equivalences determine paths in the universe of types.

We have something of the form

$$\text{types} \overset{\dashleftarrow}{\to} \infty\text{-groupoid} \overset{\simeq}{\to} \text{spaces up to weak eq.}$$

To see why univalence is a rather *strong* assumption, consider that one half of the statement is essentially the claim that "isomorphic objects are identical". Of course working mathematicians regularly conflate isomorphism with identity, but we secretly know that two isomorphic objects can stand for very different conceptual entities (e.g. an abstract group and a permutation group). To say that identity and isomorphism are actually *the same* seems, on one level, to be an extraordinary loss.

To see why univalence is a rather *natural* assumption, consider that we are really expanding the notion of identity (rather than collapsing the notion of equivalence) so that it has a finer notion of structure [7]. In effect, this means that we ignore non-structural notions of equality. All mathematical objects are to be regarded only in terms of their structure—no more secrets.

Perhaps most importantly from a practical perspective, univalence makes it much easier to define and work with *higher inductive types* (like circles, pullbacks, smash products, higher homotopy groups, cohomology, etc.), which then open up a vast array of type-theoretic approaches to homotopy theory and homotopy-theoretic approaches to type theory.

The chief reference is the HoTT book [112] written at the Institute for Advanced Study in 2013.

C.1 Very brief review of type theory

Given types A, B, C one can write $a : A$ to mean that a is a term of type A. This is a declaration or *judgment*, as opposed to a proposition. One can then form types $A \times B$, A^B, $A + B$, as well as function types $A \to B$ (so $f : A \to B$ means that f is a term of the function type $A \to B$). All these types live a universe of types $\mathcal{U}$, which is not a type for reasons going back to Russell's paradox.

A *dependent function type* or *dependent product* $\prod_{x:A} B(x)$ is a function $B : A \to \mathcal{U}$ that returns types in the universe of types $\mathcal{U}$. So morally it's a family of types. In a HoTT-flavored proof the dependent function will often be read as "for all $x : A$, $B(x)$ is true", so I like to call it the "*constructive* $\forall$": not only is $B(x)$ true, but we can give you the function demonstrating it for every $B(x)$.

Its dual, the *dependent pair* or *dependent coproduct* $\sum_{x:A} B(x)$, is also a family of types modeled on a function $B : A \to \mathcal{U}$, except in this case elements are not functions $B(x)$ but pairs of the form (a, b) for $a : A$ and $b : B(a)$. Since the dependent pair gives us the specific element in $B(a)$ that "satisfies" B, we can think of it as a "constructive $\exists$".

Given $a, b : A$, we have the *identity type* $Id_A(a, b)$, where each element $p : Id_A(a, b)$ represents a proposition "$a = b$"—in fact, a *proof* of "$a = b$". (The $=$ symbol, sometimes written $=_A$, is better understood as "is similar according to A".) Moreover, we have a stack of identity types: given $p, q : Id_A(a, b)$, we have another type $Id_{Id_A(a,b)}(p, q)$. Whereas these types were among the more mysterious entities in vanilla Martin-Löf type theory, in homotopy type theory these have a direct interpretation as the (higher) path objects of A. The original contribution of HoTT lies in its interpretation of such identity types (more on this later), which leads directly to the motto: "types are ∞-groupoids".

As is plain from the above definitions, homotopy type theory is intimately connected to logic, and it can be quite fun (and enlightening) to see that all our magical constructions in topology have such plain logical realizations. For me, it was a little shocking to realize that a "predicate" $B(x)$ on A could be read as a fibration (where the path lifting property of a fibration corresponds to the ability to "transport" predicates between identical terms in A) and that $\prod_{x:A} B(x)$ and $\sum_{x:A} B(x)$ could be read, respectively, as the space of sections and the total space of the fibration.

Example: S^1

The existence of higher path structure can be exploited in the construction of higher inductive types, which intuitively are types where we specify generators not only for the terms but also for the higher paths. This can give us nice, synthetic definitions. For example, the circle S^1 is generated by a point $* \in A$ and one nontrivial path $p : Id_A(*, *)$—in particular, all higher paths are trivial, so there is no higher homotopy, as expected.

C.2 Propositions as (some) types

A type P is a *mere proposition* if any two elements of P are equal, e.g. the type $Id_P(x, y)$ is inhabited for all $x, y : P$. In other words, P, is (-1)-connected; the idea is that P is true if and only if it is inhabited. Mere propositions are also called *subterminal objects* or h-propositions; the language suggests that while *any* type can be regarded as a proposition (through a process called propositional truncation $|| \cdot ||_{-1}$ or (-1)-truncation, which allows us to recover traditional logic as a special "trivial case" of the type theory), in general there is quite a bit more going on.

For example, a type S is a *set* (or a 0-type) if for any terms $x, y : S$, $Id_S(x, y)$ is a proposition. In other words, S is path-connected. There is a corresponding notion of 0-truncation for sets.

We can keep going up to obtain an inductive hierarchy of n-connected types, where a type A is an *n-type* if for all $x, y : A$ we have $id_A(x, y)$ is an $n-1$-type. So a (mere) proposition is a -1-groupoid, a set is a 0-groupoid, a normal groupoid is a 1-groupoid, and so on. For each n-type there is a corresponding notion of n-truncation, denoted $|| \cdot ||_n$, that forgets about or equates all the higher path structure above level n.

Crucially, such n-truncation should be thought of as a *method of abstraction*. Types are already abstract entities in a strong sense, but what makes them useful and powerful is that we can articulate a clean way of comparing our abstractions "on the same level".

C.3 Univalent foundations

At its core, type theory is based on the idea that we can move from an abstract entity (given by its essential i.e. universal properties) to instances; the instances are *representations of the types*[52] where to be an abstract entity is to have many individual representatives that can "do the work". Identity types make perfect

[52] So this goes in the reverse direction of the usual in learning, in which we construct abstract representations of objects in the data.

sense in this framework, since we are considering the relationships between different individuals. By contrast, ZF(C) and other extensional frameworks hide the abstract nature of mathematical concepts.

But as Marquis [70] noted, abstractness for mathematicians is more of a *epistemological* characterization, rather than an *ontological* characterization. Mathematical abstraction follows a principle of invariance: it is based on extracting some criterion of identity: isomorphisms for groups, isometry for metric spaces, equivalences for categories. There is often a temporal delay in this criterion of identity, e.g. Hausdorff presents topological spaces in 1914 and 10 years later Kuratowski defines homeomorphisms, but the point is that there is some sort of method for constructing and characterizing the abstract entities, usually some sort of invariant method.[53] Groups, rings, fields, metric spaces, topological spaces, Banach spaces, categories, etc. are all abstract entities in this sense. A type X, if we take the entire head-to-toe ∞-groupoid picture of the type, is just an abstract entity in this sense: something built out of a far more powerful criterion of identity that allows comparison on many different levels of identity (by n-truncation), not merely via set-theoretic predicates.

The univalence axiom is precisely such a principle of abstraction. Recall that univalence states that

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B).$$

To say that identity is equivalent to equivalence is to say that once you have abstracted properly, it is possible to identify what was previously seen as being different.

Question 30. Work in HoTT has obvious implications to homotopy theory and type theory, but there is another direction we can take it. As Voevodsky suggested [**?**], one of the aims of the univalent foundations program is to develop a "natural language" of mathematics which would be easily computer-checkable, thus allowing mathematicians to guarantee the correctness of their proofs. Unfortunately, existing proof assistants like Agda are far away from being practically useful except to specialists in formal mathematics; coding proofs of even "trivial" propositions like $2 \simeq \text{Aut}(2)$ is incredibly tedious. (That said, I have no personal experience with Coq, the other major proof assistant in use.) How can they be made more useful; how can we design a "proof assistant" that is as indispensable to mathematical practice as, for example, LaTeX? Is formal verification even the right value proposition? Clearly HoTT has some answers, but not all of them. These questions broach a much deeper subject, including what I think is simultaneously the most important, most interesting, and most overlooked question in the foundations: how do mathematicians (i.e. people) actually *do* mathematics?—where do conjectures come from?

[53] Wollheim uses types to construct artistic ontologies in much the same way: what is a book? What is a painting? It comes down to some criterion of comparison!—along with a "construction principle."

D Topological data analysis

Given a finite metric space X, one way of clustering points in X is to define a parameter ϵ such that $\{x_0, x_1, ..., x_n\} = A$ are in a cluster if $d(x_i, x_j) < \epsilon$ for all $x_i, x_j \in A$. (This is called single-linkage clustering.) Then the *0-th persistent homology* of X is a summary of the clustering behavior under *all* possible values of ϵ at once. Suppose that members of X are drawn from some manifold M according to some distribution; then the persistent homology tells us something about the shape and connectivity of M which is robust to noise.

Persistent homology is the main tool in *topological data analysis* (TDA), a discipline of applied topology which seeks to use topological tools like homotopy and homology to extract qualitative geometric information from large, high-dimensional data sets. The coarseness of the invariants is thought to be an advantage in such data, where often we do not have a good interpretation of the metric or the basis.

There are others tools in TDA, some of which we will review. Singh's *Mapper* algorithm [100], for example, is a method of "partial clustering" based on the homotopy colimit construction. Blumberg and Mandell [12] describe an approach to quantitative homotopy theory that applies persistent homology to the *contiguity complex* (understood as some approximation of $[X, Y]$) of maps between two simplicial complexes.

D.1 Persistent homology

The following definitions are slightly modified from [19].

Definition D.1. Let $\mathcal{C}$ be any category, and $\mathcal{P}$ be a partially-ordered set. We regard $\mathcal{P}$ as a category in the usual way, i.e. with object set $\mathcal{P}$ and a unique morphism from x to y whenever $x \leq y$. Then a *$\mathcal{P}$-persistence object* in $\mathcal{C}$ is a functor $\Phi : \mathcal{P} \to \mathcal{C}$.

We denote the category of $\mathcal{P}$-persistence objects in $\mathcal{C}$ by $\mathcal{P}_{pers}(\mathcal{C})$.

The key technical and practical example is that of $\mathbb{N}$-persistence simplicial complexes, which are functors $\Phi : \mathbb{N} \to \mathbf{SimpComplex}$. Given a finite metric space X (representing our data) and any subset $V \subseteq X$, consider the Cech complex $C(V, \epsilon)$ for $\epsilon \in \mathbb{N}$, e.g. the nerve of the covering obtained by open balls of radius ϵ. Since $C(V, \epsilon) \subseteq C(V, \epsilon')$ for $\epsilon \leq \epsilon'$, this defines an $\mathbb{N}$-persistence simplicial complex, which we may regard as a filtration K of simplicial complexes. Passing to the associated chain complexes gives us an $\mathbb{N}$-persistence chain complex.

Intuitively, the idea of persistent homology is to consider the simplicial inclusions $K^i \to K^{i+1}$ and the corresponding image of $H_*(K^i)$ in $H_*(K^{i+1})$; this will give us the desired behavior of homology as we vary ϵ. The actual computation of the persistent homology will return a set of intervals corresponding to the appearance and disappearance of simplices as ϵ increases in (integral) value from 0.

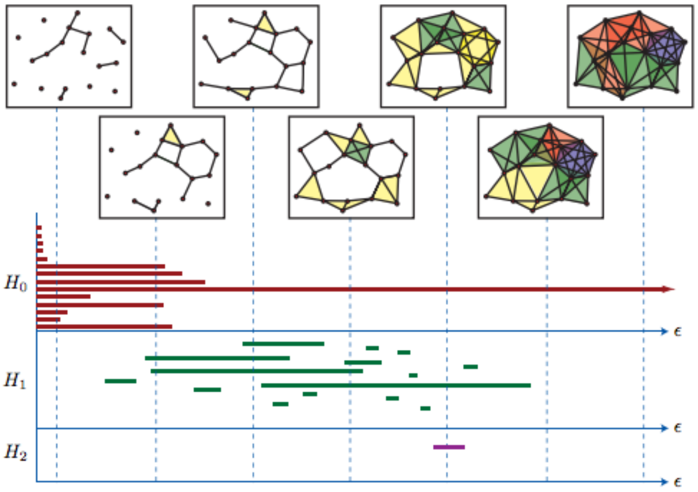


Figure D.1: The persistence barcode of a complex. Image copied from [41].

Denote the ϵ-persistent k-th homology of the i-th simplicial complex K_i as

$$H_k^{i,\epsilon} := Z_k^i/(B_k^{i+\epsilon} \cap Z_k^i).$$

This is well-defined since both $B_k^{i+\epsilon}$ and Z_k^i are subgroups of $C_k^{i+\epsilon}$, so their intersection is a subgroup of Z_k^i. For computational reasons we do not have space for here [123], we will always use field coefficients, so the persistent homology groups are actually F-vector spaces, and may be computed by standard means [31]. To obtain the desired summary, we notice that an $\mathbb{N}$-persistence F-vector space can be understood as a (non-negatively) graded $F[t]$-module, which has a nice structure theorem:

Theorem D.2. *Let M_* denote any finitely-generated non-negatively graded $F[t]$-module. Then there are integers $\{i_1, ..., i_m\}, \{j_1, ..., j_n\}, \{l_1, ..., l_n\}$ such that*

$$M_* \cong \bigoplus_{s=1}^{m} F[t](i_s) \oplus \bigoplus (F[t]/t^{l_t})(j_t)$$

Thus we can associate to each graded $F[t]$-module a set of ordered pairs (i, j) with $0 \leq i < j \in (\mathbb{Z} \cup \infty)$—the interval (i, j) describes a basis element for the homology starting at $\epsilon = i$ and ending at time $\epsilon = j - 1$. The set of such intervals will be finite since our graded $F[t]$-module is finitely-generated, since our original simplicial complex was finite.

There are ways of extending the theory to $\mathbb{R}$-persistence objects via certain mappings from $\mathbb{N}$ to $\mathbb{R}$, or of replacing Cech complexes with Vietoris-Rip complexes or weak witness complexes (which have a built-in notion of approximation). Their theory is not substantially more interesting than that of $\mathbb{N}$-persistence, so we will not review them here.

Question 31. Is there much higher-dimensional topological structure in actual scientific data? Even if yes, it's difficult to interpret the meaning of that high-dimensional structure.

Instead of asking whether it or not it is there, we might instead talk about how to promote or look for situations where there is likely to be higher-dimensional structure. We can talk of a "phase transition". In percolation theory, the main result is the existence of a "critical probability" at which the problem flips from a situation of no-paths to one of many-paths. This result is not well understood in higher dimensions—in particular, does Poincaré duality still hold?

D.2 Persistent cohomology

Notes from Perea's talk on April 22 [**?**]: persistent cohomology is about using the shape of data to choose a good coordinate scheme. First, observe representability of cohomology, e.g. $H^1(X;\mathbb{Z}) = [X, S^1]$. So by taking a cohomology class, we have a map from our data into S^1. What about projective coordinates? E.g. $H^1(X;\mathbb{Z}_2) = [X, \mathbb{R}P^\infty] \to_{i^*} [X^{(1)}, \mathbb{R}P^2$ for $i : X \to X^{(1)}$? The i^* is the dimensionality reduction to projective coordinates (recall, PCA works on projective space)!

Examples $H^1(\mathbb{R}P^2;\mathbb{Z}_2)$.

Example: $H^1(T;\mathbb{Z}_2)$.

Upshot: every time you have a cohomology class on your data in dimension 1 to $\mathbb{Z}_2$, you get a map from that data into RP^∞, e.g. projective coordinates, then you can do dimensionality reduction.

Example: images of a line in a white box, parameterized by θ and r. [Boundary detection... seems very similar to Gunnar Carlsson's example in the AMS?] This type of data sets benefits from being put into projective coordinates.

Bad thing about Brown Representability: tends to collapse all your data to 0 (bad if you want to do data analysis). Way you fix it: consider the "harmonic cocycle" (see work by ...). The harmonic representative tends to "spread out" the points a bit better. Use of line bundles: better able to spread out the points for computational purposes. Having F line bundles is the same as having cohomology classes with sheaf cohomology coefficients. Point is the cohomology is something we can compute, gives line bundles, and line bundles have better properties when we write down the classifying maps (see Perea's diagram). It's a better map and in the end they're the same map. (Examples: real and complex case: story in terms of transition functions, change into story about an exact sequence of sheaves ($\mathbb{Z} \to \mathcal{C} \to \mathcal{C}^*$) called the exponential sequence.)

Upshot: shape-aware dimensionality reduction.

D.3 Mapper

Recall that clustering is the finite case of finding connected components. In topology, one general approach to finding connected components is to analyze quotients of spaces as relations in a larger space. For example, instead of studying X/A, one enlarges the space X to a larger space $X' = X \cup CA$ in which A

is contractible, where CA is the cone over A attached at A. Then the quotient X/A is $X - A \cup *$ (the vertex of CA), while the 'transition' part CA encodes the topology of the quotient.

This transition part is important to keep track of, since it will often happen that we want to use homotopy types to build a space, but our gluing and quotient constructions depend on features of spaces which are not invariant under homotopy equivalence. For example, we can use two disks to construct a sphere, but clearly two points cannot be glued to make a sphere, even though the disks are contractible. The problem is that the quotient construction where we glue the boundary of the disks depends on having an inclusion (more precisely, it needs a cofibration; something with the homotopy extension property).

The moral is that you want to fix constructions like gluings and quotients by better ones which depend only on the homotopy type of the space, making it much easier to study the resulting space. This is just the idea behind a homotopy colimit, which "glues along homotopies".

Definition D.3. The homotopy colimit of a (gluing) diagram $D : J \to \mathbf{Top}$ is the geometric realization of its simplicial replacement. That is,

$$\text{hocolim } D = |\text{srep}(D)|$$

Sometimes we will write $\text{hocolim}_J D$ to remind us of the indexing category.

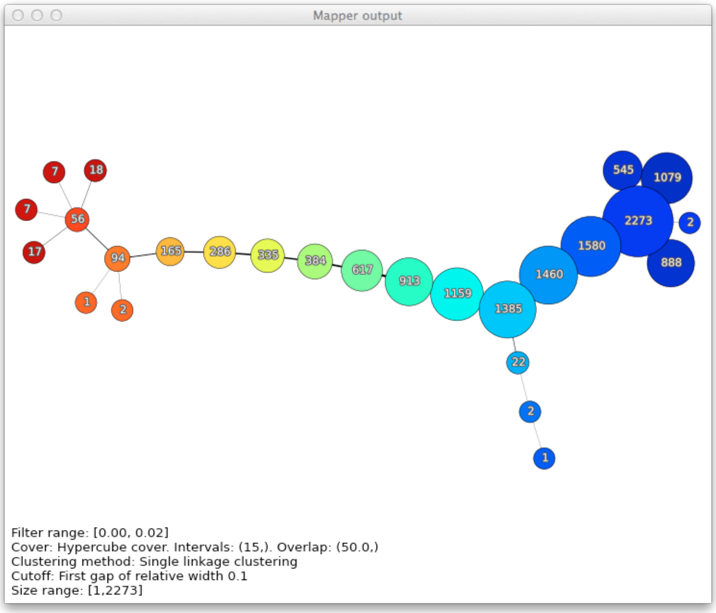
Recall that the simplicial replacement of a gluing diagram D is a simplicial complex whose simplices track how the various spaces get glued; vertices are the spaces, edges are the morphisms, faces are compositions of two morphisms, 3-simplices are compositions of three morphisms, and so on, while the face and degeneracy maps are obtained by indexing the morphisms and identifying simplices by their common indices.

Simplicial replacements of diagrams in **Top** are, essentially, special cases of nerves, which construct simplicial sets from a diagram in *any* small category. This is the reason that Stovner [105] describes Singh's Mapper algorithm [100] as a homotopy colimit, since the topological content of Mapper comes down to constructing the nerve of a diagram in **TopCov** (the category of topological spaces with associated coverings). Specifically,

$$\text{Mapper}(f^{-1}(\mathcal{U})) := N(\mathfrak{C}(f^{-1}(\mathcal{U})))$$

where $f : X \to \mathbb{R}^n$ is a filter function, $\mathcal{U}$ is a cover of $\mathbb{R}^n$, and $\mathfrak{C}$ indicates an arbitrary clustering method (this can be thought of as a way of passing to a subcover).

In actual experiments, the filter function is often a function to $\mathbb{R}$ like the k-nearest neighbor distance, the eccentricity, or the graph Laplacian, all of which give some sort of geometric characterization of the point. The choice of cover is even important, and is usually determined according to two parameters: the typical size of a set in the cover and the desired percent-overlap between sets in the cover. The choice of clustering method is usually insignificant.



```
Filter range: [0.00, 0.02]
Cover: Hypercube cover. Intervals: (15,). Overlap: (50.0,)
Clustering method: Single linkage clustering
Cutoff: First gap of relative width 0.1
Size range: [1,2273]
```

Figure D.2: Partial cluster using a 2-nearest-neighbor distance filter on 3D data, grabbed from Python Mapper [**?**]. Can you guess what the original shape was? (Hint: it rhymes with 'bat'.)

Functorial clustering

The following definition is from [19].

Definition D.4. A clustering algorithm is *functorial* if whenever one has an inclusion $X \to Y$ of point clouds, i.e. a set-map preserving distances, then the image of each cluster constructed in X under f is included in one of the clusters in Y, so that we have an induced map of sets from the clusters in X to the clusters in Y.

The clustering method used by Mapper is functorial in the sense above, since any clusters formed by the clustering method will be contained in the clusters that would have been formed had we applied the standard clustering method to the original (unfiltered) data.

However, the underlying idea of functorial clustering has applications far outside Mapper. Carlsson suggests a link from functorial clustering (think of the functor $\pi_0(X)$ from **Top** to **Set**) to étale homotopy theory, and suggests several related definitions of clustering functors by varying the morphisms of their domain, the category of finite metric spaces.

D.4 Quantitative homotopy

TBD. Develop the connection to spectra?

> "It now seems clear that the way to investigate the subtleties of low-dimensional manifolds is to associate to them suitable infinite-dimensional manifolds (e.g. spaces of connections) and to study these by standard linear methods (homology, etc.)." - Atiyah

E TQFT

This section summarizes some ongoing discussions with Yiannis Vassopoulos on applications of topological quantum field theories (TQFT) to neural networks.

E.1 Very brief review of TQFT

The following definition is from Atiyah [6].

Definition E.1. A topological quantum field theory (TQFT), in dimension n over a ground ring Λ, consists of the following data: (1) a finitely-generated Λ-module $Z(\Sigma)$ associated to each oriented closed smooth n-dimensional manifold Σ and (2) an element $Z(M) \in Z(\partial M)$ associated to each oriented smooth $(n+1)$-dimensional manifold (with boundary) M, subject to the following axioms:

1. Z is *functorial* with respect to orientation preserving diffeomorphisms of Σ and M,

2. Z is *involutory*, i.e. $Z(\Sigma^*) = Z(\Sigma)^*$ where Σ^* is Σ with opposite orientation and $Z(\Sigma)^*$ denotes the dual module,

3. Z is *multiplicative*, i.e. $Z(\Sigma_1 \cup \Sigma_2) = Z(\Sigma_1) \otimes Z(\Sigma_2)$ for Σ_1 and Σ_2 disjoint.

Remark E.2. The multiplicative axiom (3) should be compared to the additive axiom of a homology theory, i.e. $H(X_1 \cup X_2) = H(X_1) \oplus H(x_2)$ for disjoint spaces. It says, in essence, that $Z(M)$ can be computed in many different ways, by "cutting M in half" along any Σ.

In the context of QFTs, n is usually ≤ 4.

To appreciate what is going into this definition, it helps to realize that our n-dimensional TQFT is trying to represent (in the sense of a functor to **Vect**) a hom-category **nCob** of $(n+1)$-dimensional oriented cobordisms between n-dimensional manifolds [8]. The axioms above follow directly from the structure of **nCob**. While the duality structure is important from an algebraic perspective, the higher geometry (what [8] calls the symmetric braiding) is determined by the multiplicative structure, which articulates the rule for how to decompose the dynamics of the cobordism, i.e. break it up along its 'time' dimension, compute and verify the parts, then study their composition through some suitable inner product structure that lifts to **Vect**.

So the dynamics are embedded in the hom-spaces of **nCob**. Conveniently, our natural description of this hom-space is recursive, e.g. each object in **nCob** gives a morphism in **(n-1)Cob**.[54] A full algebraic picture of **nCob** should collate all the data of these n-morphisms, which strongly implies that the natural setting for TQFT should be n-categorical. Further, since the composition in **nCob** is associative up to homotopy, we can pass to A_∞-categories, where the associativity axiom for morphisms, $f \circ (h \circ g) = (f \circ h) \circ g$, is relaxed "up to higher coherent homotopy". That is, we regard the axiom as true so long as we can homotope one of f, g, h to maps satisfying the axiom. In a more restricted but more obviously convenient sense for TQFT, (linear) A_∞-categories are settings in which we can study the homology of the hom-space between objects—in this case, we assume that the hom-spaces have the structure of a chain complex of linear spaces. So the usual A_∞-categories we will consider are categories enriched (homotopically) over a category of chain complexes. A_∞-algebras are A_∞-categories with only one object, corresponding to **1Cob**.

Question 32. In dimensions 3 and 4, we have a zoology of Kirby moves—blow-ups, handle-slides, slam-dunks—used to specify the relations (on the generators) of a TQFT. There are some standard explanations—one can point to the theorems—for the difficulty in these dimensions due to work by Smale and Thurston, but what is the n-categorical, hom-space explanation for what makes them special?

As many have observed, an n-category version of a TQFT can be analyzed as structures on certain directed graphs by interpreting the morphism diagrams of **nCob** literally. In particular, Kontsevich and Vassopoulos [59] do this by thickening a ribbon graph in order to construct a surface with boundary to which the graph is homotopic. This surface is used to construct a differentially-graded PROP (a 'products and permutations category') with a particular structure, and the Hochschild chain complex over this dg-PROP will have the structure of a TQFT.

E.2 A few questions

Some first observations, from discussion with Vassopoulos.

1. Of course (the graph of) a feedforward neural network can be thought of as an oriented 1-cobordism. In a perfectly naïve way, we can associate to vertices of the network an A_∞-algebra structure (specified by their incident edges) and "do TQFT". It's unclear how helpful this is, however.

2. Almost all neural networks learn by some form of convex optimization, so it's very natural to consider the *error (hyper)surface* of the neural network as a way of analyzing the function. Recall that this hypersurface is determined by the data X, the graph structure of the neural net, and the particular loss function / optimization method. Following our strategy for **nCob**, we can study the local optima of this hypersurface using

[54] In one manner of speaking, all TQFTs are built around this description, just as all homology theories are built around the recursive description of complexes.

Morse theory, and generate a category with objects the critical points and morphisms the flow lines. In principle, it should be possible to use this categorical description to give some algebro-topological structure on the function space of all neural networks.

3. Recall the standard tree operad, which can be viewed as an algebraic structure denoting ways to expand and compress the tree by 'pasting in' simpler and subtrees (in a way that respects associativity of the pasting operation). Neural networks can be represented by operads if we think of each layer as a node (instead of each neuron).

Question 33. The function space of all neural networks can be thought as the concept space associated to a data space X, since these functions give classifications of X. What are we doing when we introduce what is essentially a homological structure (contingent on some choice of a finite sample set $S \subset X$) on the concept space? Further, can we give a 'tensorial' description of this concept space, e.g. one without the "prior geometry of assumptions about data"? This description would be something like a *knowledge representation*, a big picture way of relating concepts with data. [This is all highly speculative. Does any of this language even make sense?]

Question 34. If it is possible to study neural networks by TQFTs, can this also lead to an axiomatization of a class of learning algorithms whose internal structure is encoded by graphs? One supposes that the brain's own "algorithm", if we can call it that, must lie somewhere in this class.

Question 35. Intuitively, operads offer a very convenient formalism for thinking about neurogenesis. It's not clear, however, that they say anything about the usual AI business of optimizing a loss function. Can operads help us design invariants that characterize the behavior (output states) of the network in a "qualitative" way, as we change its connectivity—i.e. as we add and subtract not only single neurons but entire graphs and clusters of neurons?

E.3 Very brief review of neural networks

See the iPython notebook [neuPROP].

[Perspectives to add: Bayesian: take a distribution over the input, compute a posterior belief of the weights [**?**], Boltzmann machine: simulated annealing on a graph of neurons and weighted connections (think of these as "gated" tubes in a thermodynamic system), quantum: hyperplanes as quantum measurement, backprop as renormalization? [**?**], operadic perceptrons.]

F Localization

Question 36. Localization is a matter of restricting attention to a certain aspect of a space; in particular, one can localize with respect to homology with the appropriate functor. What are all the different notions of localization, ordered in increasing difficulty?

We have already reviewed the idea of the localization of a ring in Section **??**.

Of X at x.

Of a ring at U.

As sheafification. Categorically, the sheafification L is a functor $L : \mathbf{PSh} \to \mathbf{Sh}$ that sends any presheaf $F : C^{\mathrm{op}} \to \mathbf{Set}$ (on a site (C, τ)) to a presheaf defined on any $U \in C$ by

$$L(F)(U) = \operatorname*{colim}_{w:\hat{U} \to j(U)} \mathbf{PSh}_C(\hat{U}, F)$$

where $j(U)$ is the representable presheaf of U, $\hat{U}$ is a sieve inclusion of a covering family over U, and w denotes a morphism of sieve inclusions from $\hat{U}$ to $j(U)$. The colimit is over all $w \in \bar{W}$, where $\bar{W}$ is the completion of U under forming small colimits in the arrow category of $\mathbf{PSh}(C)$. We call $L(F)$ the *localization of* F.

As exact functor. [Localization as an exact functor, from nLab]

In general, localization is a process of adding formal inverses to an algebraic structure. The localization of a category C at a collection W of its morphisms is if it exists the result of universally making all morphisms in W into isomorphisms.

A localization of C by W (or "at W") is a (generally large, see below) category $C[W^{-1}]$ and a functor $Q : C \to C[W^{-1}]$ such that for all $w \in W$, $Q(w)$ is an isomorphism; for any category A and any functor $F : C \to A$ such that $F(w)$ is an isomorphism for all $w \in W$, there exists a functor $F_W : C[W^{-1}] \to A$ and a natural isomorphism $F \simeq F_W \circ Q$; the map between functor categories

$$(-) \circ Q : Funct(C[W^{-1}], A) \to Funct(C, A)$$

is full and faithful for every category A.

In abelian categories. There is also a notion specialized to abelian categories?

1. Talk about the homotopy category in relation to localization?

F.1 Analogs of localization

This section is intended to prelude a collaboration with Jeff Seely and Michael Robinson on sheaf theory for neural networks.

Question 37. What would be the analog of localization in a field like machine learning? Broadly, I want to compare learning algorithms by comparing their behavior on different subsets of data.

What if I took some proposition or "learned hypothesis" of a model and attempted to sheafify it?

The intuition: that there is a correspondence from logic—in the form of formal conditions and constraints—to geometry—in the form of "consistency conditions" on data. (Perhaps this intuition would be best formalized in the theory of topoi; unfortunately I do not have the space here to explore it.)

I am interested in the foundations of AI. My original expectation, following some comments of McLarty [77], was that studying algebraic geometry would lead me to topos theory, much as Lawvere's interest in the foundations of physics and Tierney's interest in the foundations of topology had led them to (elementary) topos theory and to Grothendieck's foundations of algebraic geometry.

A concrete example / application? I think of a functional equation (deterministic equations between variables, like PDEs), a.k.a. classical causal models, as a technique for simulation, whereas a statistical model such as a Bayesian network is really something closer to a knowledge representation—a way of connecting logic with the "real world" of data. Put that way, one feeds into other: representation and simulation. They both take place inside the person's head, so they're both epistemic, in some sense. But when we do stochastic approximation of a given functional equation, the limit of all these stochastic models may not be the original functional equation, but the "action". Pearl: "Every stochastic model can be emulated by many functional relationships (with stochastic inputs), but not the other way around; functional relationships can only be approximated, as a limiting case, using stochastic models." [This note is also in 2 value.]

> "Contrary to what occurs in ordinary topology, one finds oneself confronting a disconcerting abundance of different cohomological theories. One has the distinct impression (but in a sense that remains vague) that each of these theories amount to the same thing, that they "give the same results". In order to express this intuition, of the kinship of these different cohomological theories, I formulated the notion of "motive" associated to an algebraic variety. By this term, I want to suggest that it is the "common motive" (or "common reason") behind this multitude of cohomological invariants attached to an algebraic variety, or indeed, behind all cohomological invariants that are a priori possible." - Grothendieck

G Motives

An organizing principle for a field is not merely a classification of objects (e.g. an ontology) but a means of comparing, combining, and composing tools and methods in the field. We have seen spectra, spectral sequences, and derived categories play this role in algebraic topology, as the governing principle for constructing and relating various cohomology theories. Motives are the corresponding objects in algebraic geometry, and they form a lens through which to view the rich history of the subject.

A few ways of thinking about motives:

1. the extension of the theory of spectra in algebraic topology to algebraic geometry

2. the purely structural theory of synthetic notions like points, lines, projective planes, and so on (which is supported if we think of 'motivic' equations with coefficients in algebraic cycles, as comes up in the definition of the motivic Galois group)

3. (something on Voevodsky's reference to "Grassmann's theory of forms"? In lecture.)

Our reason for studying motives is (1) to better understand cohomology in algebraic geometry, (2) its historical importance vis-a-vis the standard conjectures, and (3) to expand a model-theoretic analogy between formal languages (such as in logic and KR) and ringed objects in algebraic geometry.

The goal of this section is to see how much of cohomology in algebraic geometry may be reasonably extended from cohomology in algebraic topology, following closely the program of Voevodsky's $\mathbf{A}^1$ homotopy theory. For the most part we will follow Voevodsky and Morel's own exposition in [117] and [83], along with notes by Dugger [**?**], Levine [**?**], and Dundas [32].

G.1 The emergence of sheaf theory

Cite [**?**, **?**] for history.

Leray's invention in 1946, "faisceau", to capture variation in the fibers of a projection: definition of sheaf of modules over a topological space. Leray later described operations of sheaves (images, quotients, limits, colimits, etc.), and linked them with spectral sequences. Diverse covering notions (carapace, flow, flasque, injective) for sheaves, and followed by unification by Godement in 1958 of terminology, concepts.

Grothendieck's reconstruction of algebraic geometry in light of the Weil conjectures: the classical Galois-Dedekind picture of Spec(V) for varieties over k versus Grothendieck's picture of Spec(A) of a (commutative) ring over a scheme.

Review of SGA: Grothendieck topologies: define coverings as stable families of morphisms, and sieves (ideals of morphisms). Topology as collections of such coverings, sieves. A site is a category with a Grothendieck topology. Grothendieck topos: sheaves over a site. Reflexive subcategories.

G.2 The search for a universal cohomology theory

[Discuss the general relationship between algebraic topology and algebraic geometry before segueing into some historical exposition. Perhaps use the analogy of the simple graph versus the weighted graph?]

The problem with Grothendieck's original construction was that it was nonconstructive. Mazur [72]:

> The dream, then, is of getting a fairly usable description of the universal cohomological functor,
>
> $$V \to H(V) \in H,$$
>
> with H a very concretely described category. At its best, we might hope for a theory that carries forward the successes of the classical theory of 1-dimensional cohomology as embodied in the theory of the jacobian of curves, and as concretized by the theory of abelian varieties, to treat cohomology of all dimensions. Equally important, just as in the theory of group representations where the irreducible representations play a primal role and have their own "logic", we

might hope for a similar denouement here and study direct sum decompositions in this category of motives, relating $H(V)$ to irreducible motives, representing cohomological pieces of algebraic varieties, perhaps isolatable by correspondences, each of which might be analyzed separately.[55]

Preview of the theory of motives

There is some rather intimidating technical machinery leading up to the main theorems of today: for example, to Voevodsky's exposition of the triangulated category of mixed motives.

See Lurie's ∞-category perspective on (derived) schemes [69]. See also Toen's paper [http://arxiv.org/abs/math/0012219] on the connection between between derived algebraic geometry and algebraic topology, via the connection between dg-algebras and rational homotopy theory.

Definition G.1. Let $\mathcal{C}$ be a small category and consider the functor category $\mathcal{S}^{\mathcal{C}}$ of functors $\mathcal{C} \to \mathcal{S}$. Call a natural transformation $X \to Y \in \mathcal{S}^{\mathcal{C}}$ a *pointwise weak equivalence* if for every $c \in \mathcal{C}$ the map $X(c) \to Y(c) \in \mathcal{S}$ is a weak equivalence.

The following is Theorem 2.0.2 in [32].

Theorem G.2. *The cellular inclusions give a model category structure, called the projective structure, on $\mathcal{S}^{\mathcal{C}}$ in which a map is a cofibration if and only if it is a retract of a cellular inclusion.*

We will use the projective structure as a foundation for further exposition.
...
[Eventually, get to a discussion of these items from [**?**]:

"To summarize, let S be a Noetherian scheme of finite Krull dimension. Then we have

1. The motivic stable homotopy category $SH(S)$
2. For a ring Λ, a category $DA^1_{\Lambda(S)}$. This can be thought of as a Λ-linear version of $SH(S)$.
3. The category $DM_{\mathfrak{B}}(S)$ of Beilinson motives over S. Roughly speaking, this is the subcategory of $DA^1_{\mathbb{Q}}$ consisting of modules over the Beilinson spectrum H^* (defined below). If $S = \text{Spec}\, k$, with k a perfect field, this category is equivalent to Voevodskys triangulated category of motives (with rational coefficients), which is usually denoted $DM_{\mathbb{Q}}(k)$.

[55] Schneps' review [95] of the Grothendieck-Serre correspondence discusses this a bit: "The first mention of motives in the letters – the first ever written occurrence of the word in this context – occurs in Grothendieck's letter from August 16: "I will say that something is a 'motive' over k if it looks like the l-adic cohomology group of an algebraic scheme over k, but is considered as being independent of l, with its 'integral structure', or let us say for the moment its $\mathbb{Q}$ structure, coming from the theory of algebraic cycles."

4. The category $DM_{\mathrm{gm}}(S)$. This is the full subcategory of compact objects in $DM_{\mathfrak{B}}$, and following Voevodsky we refer to this as the subcategory of "geometric motives."]

G.3 A very brief review of SGA

The following definitions are drawn mostly from Levine's notes [32].

Definition G.3. For a commutative ring R, the *spectrum* of R, denoted $\mathrm{Spec}(R)$, is the set of all proper prime ideals of R, i.e.

$$\mathrm{Spec}(R) := \{\mathfrak{p} \subset R : \mathfrak{p} \text{ is prime }, \mathfrak{p} \neq R\}$$

As usual, we equip $\mathrm{Spec}(R) = X$ with the Zariski topology—the closed sets are the "varieties" $V(I)$ for any ideal I of R—and structure sheaf $\mathcal{O}_X$.

Example G.4. Let F be an algebraically-closed field. Then $\mathrm{Spec}(F[t])$ is called the *affine line over* F.

Definition G.5. A *scheme* is a ringed space $(X, \mathcal{O}_X)$ which is locally the spectrum of a ring. A *morphism of schemes* $f : (X, \mathcal{O}_X) \to (Y, \mathcal{O}_Y)$ is a morphism of ringed spaces which is locally of the form $(\hat{\psi}, \tilde{\psi})$ for some homomorphism of commutative rings $\psi : A \to B$.

Let $X \in \mathbf{Sch}_k$, and let $Z_n(X)$ denote the free abelian group on the closed integral subschemes W of X with $\dim_k(W) = n$. An element $\sum_i n_i W_i$ is called an *algebraic cycle* on X (of dimension n). If X is locally equi-dimensional over k, we let $Z^n(X)$ denote the free abelian group on the codimension n integral closed subschemes of X. Elements of $Z^n(X)$ are algebraic cycles on X of codimension n.

For $W = \sum_i n_i W_i \in Z_n(X)$ (or in $Z^n(X)$) with all $n_i \neq 0$, the union $\bigcup_i |W_i|$ is called the *support* of W, denoted $|W|$.

G.4 Universality

[Give definitions of correspondences, Chow groups, motives.]

In a brief note, M. Saito (quoted in [11]) delineates two general types of cohomology theories: in the first group are the Weil-type cohomology theories like singular, de Rham, l-adic, etc., while in the second group are the Deligne-type cohomology theories—Holmstrom [53] calls these Bloch-Ogus theories—these are Deligne cohomology, absolute Hodge cohomology, the absolute étale (or continuous) cohomology, and the motivic cohomology groups, i.e. higher Chow groups. These have different notions of "universal cohomology": pure motives are universal among Weil-type cohomology theories in the sense that we can always factor the cohomology through the motive. For Deligne-type cohomology theories, the motivic cohomology groups are universal, and these may be constructed from a group of morphisms in the derived category of (mixed) motives.

A multiplicative cohomology theory[56] E is *oriented* if the restriction map $E^2(\mathbb{C}\mathbf{P}^\infty) \to E^2(\mathbb{C}\mathbf{P}^1)$ is surjective. In particular, there is some element called the *complex orientation* of E that goes to $1 \in \widetilde{E}^2(\mathbb{C}\mathbf{P}^1)$ under the restriction map. We know that every oriented cohomology theory corresponds to a formal group law, that complex cobordism is universal among oriented cohomology theories, and that every formal group satisfying a condition called Landweber exactness[57] corresponds to an oriented cohomology theory.

[Expand the story about formal group laws here?]

For extraordinary cohomology theories in algebraic geometry like algebraic cobordism or algebraic K-theory, we have no easy notion of "universality" [53].

G.5 The standard conjectures

[Plan: give a full-as-possible explanation of the third of Weil's conjectures (the "Riemann hypothesis") and its connection to the standard conjectures. This will serve as one motivation to the standard conjectures. Would it also be worthwhile to discuss Beilinson's more recent exposition [**?**] relating the conjectures to motivic t-structures? Relate to Question 9.]

[56] A multiplicative cohomology theory is one where $E^*(X)$ is not only a graded abelian group but a graded ring.

[57] Let $f(x, y)$ be a formal group law and p a prime, v_i the coefficient of xp_i in $[p]_f(x) = x +_f \cdots +_f x$. If $v_0, ..., v_i$ form a regular sequence for all p and i then $f(x, y)$ is *Landweber exact*.

H Very brief review of QBism

[It's possible we can use QBism as an reference example when we start talking about generalized inference. Come back to this and fill it in from stanford+encyclopedia+healey+quantum+bayesianism.]

This section is a brief reconstruction of my notes from a talk by Chris Fuchs.

Main goal: we want a quantum Bayesian reconstruction of quantum theory (QT).[58]

QBism is related to the Bohm interpretation in the sense that it holds the experience/epistemics of the individual as fundamental. The quantum state ψ lives "in the head of the agent". A measurement device allows the agent to perform *actions* $\{E_i\}$ that measure the quantum system. The quantum system H_d is a sort of "catalyst" for the information. Other agents are also just quantum systems.

If we wipe out the agent, the quantum state goes poof! But note, H_d did not disappear. Quantum states have no "ontic hold" on the world in QB. Note: the standard interpretations hold that the probabilities of measurements are in fact constraints on the world, but in QB the states are only epistemic.

In QBism, QT is a *tool*, not a fundamental statement about reality. But as with other tools, you can study the tool in order to understand reality better. Alternately, you can think of QT as just one property of all matter.

"Running across the street" is on a plane with "quantum information experiment". Except in the latter, it's beneficial if we use the tool QT!

Can we conceive of the Born rule as an addition to *good decision-making*? Claim: the Born rule is *normative*, not descriptive! It is not a law of nature, since it involves some *freedom*; given some inputs, it does not specify a necessary output. In other words, the Born rule is more like "try your best to..." (akin to a religious maxim).

Of course, the Born rule *looks* like a description of a law of nature:

$$p(i) = \operatorname{Tr} \rho E_i.$$

How do we turn it into something normative? Well, remember that the Bayesian interpretation of probability is egocentric. Probability theory itself is only a tool that recommends you re-evaluate your decisions. Probability$(x) = 1$ does that mean that x is true in an ontic sense.

Bell inequality violations supposedly demonstrate *nonlocality in physical reality*. But QBism proposes something else! The key component of QBist reconstruction: projections. (Missing parts of the exposition here.) Fuch's favored approach: he needs a POVM that can completely specify a quantum state; and the candidate is SIC measurements. Used to define a rewritten Born rule in terms of probabilities:

$$p(D_j) = (d+1)\sum_i p(H_i)p(D_j|H_i) - 1.$$

[58] Fuchs says something about neutral and non-neutral reconstructions of QT, and how any QBist reconstruction will be non-neutral, but I don't remember what he means by neutral or non-neutral.

The consistency of this equation gives you a small class of GPTs; a “qplex”.

Fuchs’ challenge: quantum state space, instead of being $SU(n)$, is just a qplex of *maximum* Euclidean volume. (Criticism: how do you get intersubjective agreement in QBism?)

I Miscellaneous

I.1 Quantum error correction and it's relationship to

Pull up notes from Felix; read Gottesman `https://arxiv.org/abs/quant-ph/0004072` on stabilizer codes, refer to notes from rethinking workshop, plus this one from Griffiths: `http://quantum.phys.cmu.edu/QCQI/qitd213.pdf`.

I.2 Homomorphic learning

Izbicki's HLearn package [54] (the 'H' stands for homomorphic) is a machine learning library written in Haskell.[59] As Izbicki writes, HLearn's "distinguishing feature is that it exploits the algebraic properties of learning models. Every model in the library is an instance of the HomTrainer type class, which ensures that the batch trainer is a monoid homomorphism."

It's still unclear to me how a particular learning algorithm gets represented as a monoid homomorphism. My current reading of Izbicki's paper is that the monoid representation doesn't actually tell us anything interesting about the particular learning model, i.e. it does not help us characterize and differentiate learning models. The algebraic representation is really there to help us control composition and especially parallelization of learning algorithms. Many constructions in Haskell have this nice functorial character, so the result is quite believable. However, I need to study the paper more carefully; I haven't been able to actually run the accompanying code since there seems to be a bug that prevents it from compiling (last checked: mid-December 2014). Many learning models such as perceptrons, decisions trees, and a variety of clustering algorithms have already been implemented in the package.

HLearn is also connected to and complements other research on the programming structures that implement categorical approaches to probability, i.e. "probability as a monad" [**?**].

Question 38. The direction of this research is very exciting; can we eventually construct, concretely, without recourse to the voodoo magic of higher algebra, an "algebraic" category of learning algorithms?

I.3 Geometric complexity theory

In this section, we give some preliminary discussion (drawn largely from [4]) of the algebraic P vs. NP problem and Valiant's use of the permanent function to characterize the **AlgNP** class. Eventually, I would like to discuss Mulmumey's interpretation of Valiant's $\mathbf{AlgP} \neq \mathbf{AlgNP}$ conjecture in the language of geometric invariant theory.

The determinant and permanent functions play a vital role in the world of algebraic circuits, since they are complete problems for two important classes.

[59] Haskell is a functional programming language which is also the basis for Agda, the proof assistant mentioned in Section A.

(A decision problem P is said to be complete for a set of decision problems S if P is a member of S and every problem in S can be reduced to P.) To give the definition, we need the notion of *degree* of a multivariate polynomial, namely, the minimum d such that each monomial term $\prod_i x_i^{d_i}$ satisfies $\sum_i d_i \leq d$. A family of polynomials in $x_1, x_2, ..., x_n$ is *poly-bounded* if the degree is at most $O(n^c)$ for some constant $c > 0$.

Definition I.1. The class **AlgP** is the class of polynomials of polynomial degree that are computable by arithmetic formulae (using no $\div$) of polynomial size.

Definition I.2. **AlgNP** is the class of polynomials of polynomial degree that are definable as

$$f(x_1, x_2, ..., x_n) = \sum_{e \in \{0,1\}^{m-n}} g_n(x_1, x_2, ..., x_n, e_{n+1}, ..., e_m)$$

where $g_n \in \mathbf{AlgP}$ and m is polynomial in n.

Definition I.3. A function $f(x_1, ...x_n)$ is a *projection* of a function $g(y_1, y_2, ..., y_m)$ if there is a mapping σ from $\{y_1, y_2, ..., y_m\}$ to $\{0, 1, x_1, x_2, ..., x_n\}$ such that $f(x_1, x_2, ..., x_n) = g(\sigma(y_1), \sigma(y_2), ..., \sigma(y_m))$. We say that f is *projection-reducible* to g if f is a projection of g.

Theorem I.4 (Valiant)**.** *Every polynomial on n variables that is computable by a circuit of size u is projection reducible to the determinant function (over the same field) on $u + 2$ variables. Every function in **AlgNP** is projection reducible to the permanent function (over the same field).*

From MathOverflow: Mulmuley's key idea is to use symmetries to organize not the functions themselves, but to organize the algebro-geometric properties of these functions as captured by particular polynomials p; this enables the use of representation theory in attempting to find such a p.

I.4 Asynchoronous computation

In a landmark 1999 paper, Herlihy and Shavits [51] introduced a topological formalism for analyzing a particular class of problems in the field of asynchronous computation (think: multiprocessor CPUs, RAM, online networking, or most methods of parallel computation). In such problems, we start with some set of computational processes each of which starts with a private input value, and we wish to organize their interactions so that they all halt with some specified output value. However, processes may fail, or be delayed, and these failures may impede the computations of other processes. A relation $\Delta \subset \vec{I} \times \vec{O}$ of input values and output values is called a *task*, and a program that solves the task is called a *protocol.* A major part of the theory is to design a protocol that is *wait-free*: a wait-free protocol guarantees that a nonfaulty process will halt with an acceptable output value in a fixed number of steps, regardless of delays or failures by other processes.

Following Herlihy and Shavits, we assume that a computational process is something that reads and write variables to (shared) memory, usually multiple times during a protocol.

The key step in their paper was to associate to every task and to every protocol a particular simplicial complex. In the formalism, a protocol solves a task if and only if there exists a simplicial map from the protocol complex to the task complex; the main theorem states that the protocol is *wait-free* if and only if the simplicial map satisfies a relatively simple coloring condition on simplices. Specifically:

Theorem I.5 (Asynchronous Computability Theorem)**.** *A decision task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *[these are all complexes representing the task] has a wait-free protocol using read-write memory if and only if there exists a chromatic subdivision* σ *of* $\mathcal{I}$ *and a color-preserving simplicial map*

$$\mu : \sigma(\mathcal{I}) \to \mathcal{O}$$

such that for each simplex S *in* $\sigma(\mathcal{I})$*,* $\mu(S) \in \Delta(carrier(S, \mathcal{I})$*.*

In a particularly crisp and elegant result leading up to the main theorem, the authors show that every wait-free protocol corresponds to a *contractible* simplicial complex $\mathcal{I}$.

Question 39. According to the nLab, it was John Roberts who originally understood (in the context of QFT) that general cohomology is about coloring simplices in ∞-categories. What does this mean exactly, and can we apply this intuition to the coloring condition in [51]'s main theorem? If so, can we obtain some sort of generalization?

Question 40. One of the key ideas here is that just as the consequences of single failures in an asynchronous computation could be represented by graphs [**?**], the consequences of multiple failures could be represented by simplicial complexes. How does this idea accord with the idea from sample compression, that one-inclusion graphs can be replaced by simplicial complexes? Is there any deeper reason for the similarity?

Example: unique-id

For example, consider the *unique-id* task: each participating process $P_i \in \{0, ..., n\}$ has an input $x_i = 0$ and chooses an output $y_i \in \{0, ..., n\}$ such that for any pair $P_i \neq P_j$, $y_i \neq y_j$. This task has a trivial wait-free solution.

Example: fetch-increment

In this task, each participating process $P_i \in \{0, ..., n\}$ has an input $x_i = 0$ and chooses a unique output $y_i \in \{0, ..., n\}$ such that (1) for some participating index i, $y_i = 0$, and (2) for $1 \leq k \leq n$, if $y_i = k$, then, for some $j \leq i, y_j = k - 1$. This task has no wait-free solution in read/write memory if one or more processes can fail.

Question 41. Why?

References

[1] Samson Abramsky, Rui Soares Barbosa, Kohei Kishida, Raymond Lal, and Shane Mansfield. Contextuality, cohomology and paradox. In *25th EACSL Annual Conference on Computer Science Logic*, 2015.

[2] Samson Abramsky and Adam Brandenburger. The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics*, 13, 2011.

[3] Samson Abramsky, Shane Mansfield, and Rui Soares Barbosa. The cohomology of non-locality and contextuality. In Bart Jacobs, Peter Selinger, and Bas Spitters, editors, *8th Interntional Workshop on Quantum Physics and Logic (QPL 2011)*, 2011.

[4] Sanjeev Arora and Boaz Barak. *Complexity Theory: A Modern Approach.* Cambridge University Press, 2006.

[5] Michael Atiyah. How research is carried out. *Bulletin of the International Mathematical Association*, 10, 1974.

[6] Michael Atiyah. Topological quantum field theories. *Publications mathématiques de l'I.H.É.S.*, 68:175–186, 1988.

[7] Steve Awodey. Structuralism, invariance, and univalence. Technical report, Munich Center for Mathematical Philosophy, 2014.

[8] John C. Baez and James Dolan. Higher-dimensional algebra and topological quantum field theory. *Journal of Mathematical Physics*, 36:6073–6105, 1995.

[9] David Balduzzi. Distributed learning: Foundations and applications. Research statement.

[10] David Balduzzi. On the information-theoretic structure of distributed measurements. In *EPTCS*, volume 88, 2012.

[11] Luca Barbiere-Viale. A pamphlet on motivic cohomology. *Milan journal of mathematics*, 73(1):53–73, 2005.

[12] Andrew J. Blumberg and Michael A. Mandell. Quantitative homotopy theory in topological data analysis. *Foundations of Computational Mathematics*, 13(6):885–911, 2013.

[13] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[14] Valentino Braitenberg. *Vehicles: Experiments in synthetic psychology.* MIT Press, 1986.

[15] Spencer Breiner, Eswaran Subrahmanian, and Ram D. Sriram. Modeling the internet of things: A foundational approach. In *Proceedings of the Seventh International Workshop on the Web of Things*, November 2016.

[16] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.

[17] Rodney Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[18] Sebastien Bulbeck. Komlos conjecture, gaussian correlation conjecture, and a bit of machine learning.

[19] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009.

[20] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. arXiv preprint arXiv:1003.4394, 2010.

[21] Ernest Davis and Leora Morgenstern. Introduction: Progress in formal commonsense reasoning. *Artificial Intelligence*, 2004.

[22] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14:17–33, 1993.

[23] Aise Johan de Jong. Weil cohomology theories. Accessed from http://www.math.columbia.edu/~dejong/seminar/note_on_weil_cohomology.pdf, 2007.

[24] Daniel C. Dennett. *The Intentional Stance*. A Bradford Book, 1989.

[25] Thomas G. Dietterich. Learning at the knowledge level. *Machine Learning*, 1986.

[26] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, 2000.

[27] Jean Dieudonné. The historical development of algebraic geometry. *The American Mathematical Monthly*, 79(8):827–866, October 1972.

[28] Pedro Domingos. A unified bias-variance decomposition and its applications. In *ICML*, 2000.

[29] Pedro Domingos. Structured machine learning: Ten problems for the next ten years. In *Proc. of the Annual Intl. Conf. on Inductive Logic Programming*, 2007.

[30] Hubert L. Dreyfus. Why Heideggerian AI failed and how fixing it would require making it more Heideggerian. *Artificial Intelligence*, 171:1137–1160, 2007.

[31] Jean-Guillaume Dumas, Frank Heckenbach, David Saunders, and Volkmar Welker. Computing simplicial homology based on efficient Smith normal form algorithms. In *Algebra, Geometry and Software Systems*, pages 177–206. Springer, 2003.

[32] Bjorn Ian Dundas, Marc Levine, Vladimir Voevodsky, Oliver Röndigs, and Paul Arne Ostvaer. *Motivic Homotopy Theory: Lectures at a Summer School in Nordfjordeid, Norway, August 2002.* Springer-Verlag, 2002.

[33] Herbert Edelsbrunner and John Harer. Persistent homology – a survey.

[34] Andrée Charles Ehresmann and Jean-Paul Vanbremeersch. *Memory Evolutive Systems; Hierarchy, Emergence, Cognition.* Studies in Multidisciplinarity. Elsevier Science, 2007.

[35] Sally Floyd. *On Space-Bounded Learning and the Vapnik-Chervonenkis Dimension.* PhD thesis, University of California, Berkeley, 1989.

[36] Sally Floyd and Manfred K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[37] Jerry A. Fodor. Propositional attitudes. *The Monist*, 61:501–523, October 1978.

[38] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996.

[39] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 1997.

[40] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[41] Robert Ghrist. Barcodes: the persistent homology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[42] Robert Ghrist. *Elementary Applied Topology.* CreateSpace Independent Publishing Platform, 2014.

[43] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[44] Robert Goldblatt. *Topoi: The Categorical Analysis of Logic.* Dover, 2006.

[45] Odel Goldreich and Dana Ron. On universal learning algorithms. In *Impromptu Session of COLT 1996*, July 1996.

[46] John W. Gray. Fragments of the history of sheaf theory. In Michael Fourman, Christopher Mulvey, and Dana Scott, editors, *Applications of sheaves*, pages 1–79. Springer-Verlag, 1977.

[47] Owen Griffiths and A. C. Paseau. Isomorphism invariance and overgeneration. *The Bulletin of Symbolic Logic*, 22(4), December 2016.

[48] Misha Gromov. Ergostructures, ergologic and the universal learning problem: Chapters 1, 2, 3.

[49] Robin Hartshorne. *Algebraic Geometry*. Springer, 8th printing 1997 edition edition, 1977.

[50] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

[51] Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM*, 46(6):858–923, November 1999.

[52] Andreas Holmstrom. Questions and speculation on cohomology theories in arithmetic geometry. version 1. http://www.andreasholmstrom.org/research/Cohomology1.pdf, December 2007.

[53] Andreas Holmstrom. Ordinary vs generalized cohomology theories. https://homotopical.wordpress.com/2009/12/10/ordinary-vs-generalized-cohomology-theories/, December 2009.

[54] Mike Izbicki. HLearn: A Machine Learning Library for Haskell. In *Symposium on Trends in Functional Programming*, 2013.

[55] Brendan Juba. *Universal Semantic Communication*. PhD thesis, Massachusetts Institute of Technology, September 2010.

[56] Joel Kamnitzer. Algebraic geometry without prime ideals. Blog post at `https://sbseminar.wordpress.com/2009/08/06/algebraic-geometry-without-prime-ideals/`.

[57] Jyrki Kivinen and Manfred K. Warmuth. Boosting as entropy projection. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, number 134-144, 99.

[58] Daniel E. Koditschek. Dynamically dexterous robots via switched and tuned oscillators. Technical report, Artificial Intelligence Laboratory and Controls Laboratory, University of Michigan, 1101 Beal Ave, Ann Arbor, Michigan 48109-2110, December 1994.

[59] Maxim Kontsevich and Yiannis Vlassopoulos. Pre-calabi-yau algebras and topological quantum field theories. November 2014.

[60] Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2), May 2003.

[61] Dima Kuzmin and Manfred K. Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8:2047–2081, 2007.

[62] Jean Leray. Sur la forme des espaces topologiques et sur les points fixes des représentations. *Journal de Mathématiques Pures et Appliquées*, 24:95–248, 1945.

[63] Xuchun Li, Lei Wang, and Eric Sung. A study of adaboost with svm based weak learners. In *Proceedings of the International Joint Conference on Neural Networks*, 2005.

[64] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[65] Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.

[66] Rodolfo R. Llinas. *I of the Vortex: From Neurons to Self.* Bradford Books, 2002.

[67] Tom Lovering. Sheaf theory. September 2010.

[68] Jacob Lurie. *Higher Topos Theory.* Princeton University Press, 2009.

[69] Jacob Lurie. Higher algebra. 2014.

[70] Jean-Pierre Marquis. Homotopy type theory as a foundation for mathematics: some philosophical remarks. Lecture for the HoTT-NYC group, April 2014.

[71] Jon Peter May. Stable algebraic topology, 1945-1966. In I. M. James, editor, *The History of Topology.* Elsevier, 1999.

[72] Barry Mazur. What is a motive? *Notices of the AMS*, 51(10):1214–1216, 2004.

[73] John McCarthy. An example for natural language understanding and the ai problems it raises.

[74] James L. McClelland and Timothy Rogers. The parallel distributed processing approach to semantic cognition. *Nature Reviews Neuroscience*, 4:310–322, 2003.

[75] James L. McClelland and David E. Rumelhart. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition - Volume 1 (foundations).* MIT Press, 1986.

[76] Peter McCullagh. What is a statistical model? *The Annals of Statistics*, 30(5):1225–1310, 2002.

[77] Colin McLarty. The uses and abuses of the history of topos theory. *British Journal of Philosophy of Science*, 41:351–375, 1990.

[78] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Sizzerman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 2006.

[79] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, 1st edition, 1969.

[80] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning.* MIT Press, 2012.

[81] Robert C. Moore. *Logic and representation.* Center for the Study of Language and Information, 1995.

[82] Shay Moran and Manfred K. Warmuth. Labeled compression schemes for extremal classes. In *International Conference on Algorithmic Learning Theory*, pages 34–49, 2016.

[83] Fabien Morel and Vladimir Voevodsky. $\mathbf{A}^1$-homotopy theory of schemes. *Publications mathématiques de l'I.H.É.S.*, 90(1):45–143, 1999.

[84] Alan Newell. The knowledge level: Presidential address. *AI Magazine*, 2(2), 1980.

[85] Albert B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

[86] Natalya F. Noy. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33(4), 2004.

[87] Brian Osserman. The weil conjectures. In Timothy Gowers, editor, *Princeton Companion to Mathematics*, pages 729–732. Princeton University Press, 2008.

[88] Pierre-Yves Oudeyer, Adrien Baranes, and Frédéric Kaplan. Intrinsically motivated learning of real-world sensorimotor skills with developmental constraints. In Gianluca Baldassarre and Marco Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems.* Springer, 2013.

[89] Judea Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, 2009.

[90] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007.

[91] Hilary Putnam. *Mind, Matter, and Reality*, volume 2 of *Philosophical Papers*. Cambridge University Press, 1975.

[92] Michael Robinson. Sheaf and duality methods for analyzing multi-model systems. arXiv: 1604.04647v2.

[93] Benjamin I.P. Rubinstein and J. Hyam Rubinstein. A geometric approach to sample compression. *The Journal of Machine Learning Research*, 13(1):1221–1261, 2012.

[94] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2014.

[95] Leila Schneps. A biographical reading of the grothendieck-serre correspondence. *Mathematical Intelligencer*, 2007.

[96] Jean-Pierre Serre. Faisceaux algébriques cohérents. *The Annals of Mathematics*, 61(2):197–278, March 1955.

[97] Jean-Pierre Serre. *Algebraic Groups and Class Fields*. Springer, 1988.

[98] Mike Shulman. Cohomology, July 2013.

[99] Siggraph. *Evolving Virtual Creatures*. Computer Graphics, July 1994.

[100] Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In M. Botsch and R. Pajarola, editors, *Eurographics Symposium on Point-based Graphics*, 2007.

[101] David I. Spivak. *Category Theory for Scientists*. MIT Press, 2014.

[102] David I. Spivak and Robert E. Kent. Ologs: A categorical framework for knowledge representation. *PLoS ONE*, 7(1), 2012.

[103] David I. Spivak and Joshua Z. Tan. Nesting of dynamic systems and mode-dependent networks. *Journal of Complex Networks*, 5(3):389–408, July 2017.

[104] David I. Spivak, Christina Vasilakopoulou, and Patrick Schultz. Dynamical systems and sheaves. *ArXiv e-prints*, September 2016.

[105] Roar Bakken Stovner. On the mapper algorithm: A study of a new topological method for data analysis. Master's thesis, NTNU Trondheim, 2012.

[106] Richard Szeliski. Computer vision: Algorithms and applications.

[107] Joshua Tan, Christine Kendrick, Abhisheky Dubey, and Sokwoo Rhee. Indicator frameworks. In *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*, pages 19–25, 2017.

[108] Joshua Z. Tan, Andrea Censi, and David I. Spivak. A categorical theory of design. In process.

[109] E. K. Tang, P. N. Suganthan, and X. Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006.

[110] The Mathematical Society of Japan. *Encyclopedic Dictionary of Mathematics.* The MIT Press, 4th edition, 2000.

[111] Richmond Thomason. Logic and artificial intelligence. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Spring 2014.

[112] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* http://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[113] Dmitry Vagner, David I. Spivak, and Eugene Lerman. Algebras of open dynamical systems on the operad of wiring diagrams. https://arxiv.org/abs/1408.1598.

[114] Ravi Vakil. Baby algebraic geometry seminar: An algebraic proof of riemann-roch. February 2000.

[115] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[116] Andrea Vedaldi. *Invariant Representations and Learning for Computer Vision.* PhD thesis, UCLA, 2008.

[117] Vladimir Voevodsky. $\mathbf{A}^1$-homotopy theory. In *Proceedings of the International Congress of Mathematicians*, volume I, pages 579–604, Berlin, 1998.

[118] Sumio Watanabe. *Algebraic geometry and statistical learning theory.* Cambridge University Press, 2009.

[119] Charles A. Weibel. History of homological algebra. In I. M. James, editor, *The History of Topology*, pages 797–836. Elsevier, 1999.

[120] Daniel M. Wolpert, Zoubin Ghahramani, and J Randall Flanagan. Perspectives and problems in motor learning. *Trends in Cognitive Sciences*, 5(11), 2001.

[121] Michal Wozniak, Manuel Grana, and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.

[122] William Zeng and Philipp Zahn. Contextuality and the weak axiom in the theory of choice. In *International Symposium on Quantum Interaction*, volume 9535 of *Lecture Notes in Computer Science*, January 2016.

[123] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.