What is string data?

Mike Douglas, String-Data 2021

[1]CMSA Harvard/Stony Brook

December 13, 2021

Abstract

Machine learning critically depends on high quality datasets.
In a theoretical subject like string theory, we can generate datasets, but what sort of data should we generate and study?
We discuss this question from several perspectives: mathematical (generating solutions), statistical (getting representative samples), and methodological (improving access to prior work).

Computers have been valuable tools for scientists for decades.

This is more and more the case in recent years, and now progress in machine learning is changing the way we do science.

There is no better illustration of this than DeepMind's AlphaFold 2, whose astounding results on protein folding at CASP 14 Nov 2020, have transformed computational biology.

Fundamental physics has not yet been transformed, but all of its data-intensive subfields rely more and more on ML. This includes not only analysis of experimental and observational data, but also theoretical/computational subfields such as lattice gauge theory, *e.g.* see the work of the Shanahan group (MIT/IAIFI).

How can we use ML to study string theory?

Of course, the researchers at this conference have been using ML to study string theory for some time now. We will hear about new work in many directions:

- Automated and assisted discovery of laws, equations, models, symmetries, conjectures
- Search algorithms (RL, genetic algorithms, *etc.*) for solving combinatorial problems, especially for finding string vacua
- ML-inspired numerical methods (CY metrics, bootstrap, *etc.*)
- Physics/theory of ML interdisciplinary connections
- And more.

As with all work in ML, the scope and quality of the results depends directly on the size and quality of the datasets used for training. Of course, it depends on many other factors. But in this talk, let us focus on datasets and computational methods.

Of course, the researchers at this conference have been using ML to study string theory for some time now. We will hear about new work in many directions:

- Automated and assisted discovery of laws, equations, models, symmetries, conjectures
- Search algorithms (RL, genetic algorithms, *etc.*) for solving combinatorial problems, especially for finding string vacua
- ML-inspired numerical methods (CY metrics, bootstrap, *etc.*)
- Physics/theory of ML interdisciplinary connections
- And more.

As with all work in ML, the scope and quality of the results depends directly on the size and quality of the datasets used for training. Of course, it depends on many other factors. But in this talk, let us focus on datasets and computational methods.

Of course, the researchers at this conference have been using ML to study string theory for some time now. We will hear about new work in many directions:

- Automated and assisted discovery of laws, equations, models, symmetries, conjectures
- Search algorithms (RL, genetic algorithms, *etc.*) for solving combinatorial problems, especially for finding string vacua
- ML-inspired numerical methods (CY metrics, bootstrap, *etc.*)
- Physics/theory of ML interdisciplinary connections
- And more.

As with all work in ML, the scope and quality of the results depends directly on the size and quality of the datasets used for training. Of course, it depends on many other factors. But in this talk, let us focus on datasets and computational methods.

So, what is a string theory dataset ?
In principle, any collection of data derived from, used in or related to string theory and its related mathematics.
The most famous example is the Kreuzer-Skarke dataset of Calabi-Yau threefolds – more specifically reflexive polytopes which determine toric varieties which admit hypersurfaces of zero first Chern class.

Of course there is no problem to generate data related to a theory, in many ways this is easier than collecting experimental and observational data. But just as not every true logical statement can be considered part of mathematics, so too not all string data is of equal value.

So how do we decide what data is important, and how should we (individually and as a community) distribute and preserve it?

So, what is a string theory dataset ?
In principle, any collection of data derived from, used in or related to string theory and its related mathematics.
The most famous example is the Kreuzer-Skarke dataset of Calabi-Yau threefolds – more specifically reflexive polytopes which determine toric varieties which admit hypersurfaces of zero first Chern class.

Of course there is no problem to generate data related to a theory, in many ways this is easier than collecting experimental and observational data. But just as not every true logical statement can be considered part of mathematics, so too not all string data is of equal value.

So how do we decide what data is important, and how should we (individually and as a community) distribute and preserve it?

Types of datasets

Explicit vs implicit vs underdetermined:

- Explicit – a finite list of items.
- Implicit – an algorithm to generate items PLUS a distribution, or an algorithm to sample items. If finite list, can be uniform.
- Underdetermined – an infinite list of items, or a finite list with no distribution (not even uniform).

Platonic vs universal vs unmotivated:

- platonic (natural, canonical) – zero or minimal dependence on arbitrary choices. Example: n-dimensional reflexive polytopes.
- universality class – some properties do not depend on arbitrary choices. Example: the prime numbers with a sequence of measures whose limit is uniform, *e.g.* $\sum_{p \leq N}$.
- unmotivated choices.

Implementation/social/methodological distinctions:
public maintained/public unmaintained/private, verified/unverified, choice of platform, *etc.*

Types of datasets

Explicit vs implicit vs underdetermined:

- Explicit – a finite list of items.
- Implicit – an algorithm to generate items PLUS a distribution, or an algorithm to sample items. If finite list, can be uniform.
- Underdetermined – an infinite list of items, or a finite list with no distribution (not even uniform).

Platonic vs universal vs unmotivated:

- platonic (natural, canonical) – zero or minimal dependence on arbitrary choices. Example: n-dimensional reflexive polytopes.
- universality class – some properties do not depend on arbitrary choices. Example: the prime numbers with a sequence of measures whose limit is uniform, *e.g.* $\sum_{p \leq N}$.
- unmotivated choices.

Implementation/social/methodological distinctions: public maintained/public unmaintained/private, verified/unverified, choice of platform, *etc.*

Types of datasets

Explicit vs implicit vs underdetermined:

- Explicit – a finite list of items.
- Implicit – an algorithm to generate items PLUS a distribution, or an algorithm to sample items. If finite list, can be uniform.
- Underdetermined – an infinite list of items, or a finite list with no distribution (not even uniform).

Platonic vs universal vs unmotivated:

- platonic (natural, canonical) – zero or minimal dependence on arbitrary choices. Example: n-dimensional reflexive polytopes.
- universality class – some properties do not depend on arbitrary choices. Example: the prime numbers with a sequence of measures whose limit is uniform, *e.g.* $\sum_{p \leq N}$.
- unmotivated choices.

Implementation/social/methodological distinctions:
public maintained/public unmaintained/private, verified/unverified, choice of platform, *etc.*

Group theory data

To get started, I looked around for relevant mathematical datasets. In particular, group theory is of central importance in theoretical physics and string theory. Since group theory has been around much longer than string theory and has many more users, it stands to reason that the issues should have been carefully considered and best practices established. On the other hand the classification problem, while important for group theory, is arguably more central for other fields such as topology and especially knot theory.

Eventually we should judge the options based on our applications in string theory, but here are some general requirements:

- Easy to learn, and interfaces with the software we are already using. For significant ML projects this generally means, easily callable from Python.
- Should contain the most common examples (small groups, infinite series) and, given the nature of string theory, the "exceptional" cases: in particular the 26 sporadic simple groups.

Group theory data

To get started, I looked around for relevant mathematical datasets. In particular, group theory is of central importance in theoretical physics and string theory. Since group theory has been around much longer than string theory and has many more users, it stands to reason that the issues should have been carefully considered and best practices established. On the other hand the classification problem, while important for group theory, is arguably more central for other fields such as topology and especially knot theory.
Eventually we should judge the options based on our applications in string theory, but here are some general requirements:

- Easy to learn, and interfaces with the software we are already using. For significant ML projects this generally means, easily callable from Python.
- Should contain the most common examples (small groups, infinite series) and, given the nature of string theory, the “exceptional” cases: in particular the 26 sporadic simple groups.

Initial observations:

- The simplest “one stop shopping” approach is to look at the two best developed computer algebra platforms, Mathematica and SageMath. Both are maintained, Mathematica more actively but not open source, SageMath is open source.
- Mathematica has a large package for permutation groups, and a nice collection of named groups including the sporadics. Also see the Knot Atlas and Mathematica package KnotTheory.
- SageMath is built on Python and includes the very sophisticated GAP (Groups, Algorithms and Programming) system.
- Among the many GAP packages is an interface to the ATLAS of Finite Group Representations, with many datasets including the sporadic groups and representations (but nobody put a Monster group representation online).
- Once one broadens the search, there is a bewildering variety of mathematical software packages – a list of almost 2000 can be found at `https://swmath.org/` .

My conclusion from this initial search is, while there is useful software and data for group theory, much remains to be done in terms of systematizing mathematical datasets, establishing best practices, and developing an organizational framework which can accept, select, validate, publish and maintain contributions.

And this is for a mature field of mathematics. For string data, we are still in early days. Basic questions such as the definition of string vacua and the questions we want to ask about them, remain open.

Still, we are already a sizable community, and we believe that this work will have lasting value, so let us discuss how to think about it and organize it.

My conclusion from this initial search is, while there is useful software and data for group theory, much remains to be done in terms of systematizing mathematical datasets, establishing best practices, and developing an organizational framework which can accept, select, validate, publish and maintain contributions.

And this is for a mature field of mathematics. For string data, we are still in early days. Basic questions such as the definition of string vacua and the questions we want to ask about them, remain open.

Still, we are already a sizable community, and we believe that this work will have lasting value, so let us discuss how to think about it and organize it.

Approximate and asymptotic distributions

In statistics, one gains tremendous conceptual and practical advantages by studying not only data, but also well-chosen model probability distributions such as normal (Gaussian), binomial, Poisson, *etc.* Although real world datasets are never exactly described by such distributions, often they give a good first approximation, sometimes for simple reasons (CLT/normal, independence/Poisson), sometimes for more subtle reasons (critical phenomena/power laws).

As theoretical physicists, we are very familiar with the use of approximations to gain intuition. A good approximation to the data distribution is also useful to design experiments, and to clean data (*e.g.* recognizing outliers).

There are many results for approximate and asymptotic distributions of mathematical datasets.

- Prime number theorem, first Hardy-Littlewood conjecture – the number of twin primes less than x is $\pi_2(x) \sim 2C_2 x/(\ln x)^2$.
- Pyber's theorem – the number gnu(n) of finite groups of order n is bounded by

$$\text{gnu}(n) \leq n^{\frac{2}{27}\mu(n)^2+\mathcal{O}(\mu(n)^{5/3})},$$

  where $\mu(n)$ is the highest power to which any prime divides n.
- Distribution of ranks of elliptic curves. In Bhargava *et al* `arXiv:1304.3971`, a discrete probability distribution $\mathcal{Q}$ is defined and conjectured to equal this distribution. This would imply that 50% of curves have rank 0, 50% have rank 1, and 0% (density) have rank ≥ 2. A refined model (Park *et al* 1602.01431) implies that only a finite number have rank > 21.

Standard approach to string compactification

- Choose a construction, *e.g.* heterotic on CY_3, type IIa/IIb on CY_3, M on G_2, F on CY_4.
- Choose compact manifold and fixed data (*e.g.* orientifolding).
- Choose auxiliary structures (bundles, branes) consistent with anomalies/tadpoles/topological conditions.
- Make discrete choices (fluxes).
- Determine effective potential on pseudo-moduli space, and find its minima.
- Check further consistency conditions (non-moduli tachyons, nonperturbative instabilities).
- Compute 4d effective Lagrangian and physical predictions.

A sequence of choices which to some extent combine combinatorially, with checks and computations becoming progressively more difficult.

Statistics of vacua

In hep-th/0303194, I began the study of approximate distributions of observables of string vacua. With Ashok, Denef and others we got many analytic results for distributions of flux vacua, from which the most important conclusions were:

- Good arguments for the Bousso-Polchinski estimate $N \sim (10-100)^{\text{Betti number}}$, which can be systematically improved to take into account results on Calabi-Yau moduli spaces, dualities, *etc.*.
- The cosmological constant Λ is uniformly distributed over a range which is **not** determined by the supersymmetry breaking scale, but rather by higher dimensional/stringy scales. This is because of the $-3|W|^2$ term. Thus the anthropic solution to the c.c. problem **does not** favor low energy supersymmetry.

This was a major input into arguments that string theory makes no clear prediction for or against supersymmetry at LHC (see 1204.6626).

Looking at more general distributions (say of matter content), the simplest idea put forward in 0303194 was that in the absence of other evidence, one should hypothesize that different observables are **independent**. For example, the problem of matching the SM could be independent of getting realistic inflation. Even so, one can still make predictions based on statistics and counting.

A few highlights of subsequent work:

- Dine and collaborators pointed out that symmetries are disfavored in a landscape and under the uniform measure. For example, preserving a symmetry in flux vacua requires setting many components of the flux to zero, reducing K^B to $K^{B_{invariant}}$.
- Gmeiner *et al* 0510170 found that about "one in a billion" brane models realize a 3 generation Standard Model.
- Marsh *et al* 1112.3034 developed a simple flux ensemble in which high scale susy breaking is disfavored.
- Taylor and Wang 1511.03209 found an F theory with (apparently) $10^{272,000}$ vacua. Many suspect that these are not real, perhaps because not all moduli are stabilized (see Bena *et al* 2010.10519).

Looking at more general distributions (say of matter content), the simplest idea put forward in 0303194 was that in the absence of other evidence, one should hypothesize that different observables are **independent**. For example, the problem of matching the SM could be independent of getting realistic inflation. Even so, one can still make predictions based on statistics and counting.

A few highlights of subsequent work:

- Dine and collaborators pointed out that symmetries are disfavored in a landscape and under the uniform measure. For example, preserving a symmetry in flux vacua requires setting many components of the flux to zero, reducing K^B to $K^{B_{invariant}}$.
- Gmeiner *et al* 0510170 found that about "one in a billion" brane models realize a 3 generation Standard Model.
- Marsh *et al* 1112.3034 developed a simple flux ensemble in which high scale susy breaking is disfavored.
- Taylor and Wang 1511.03209 found an F theory with (apparently) $10^{272,000}$ vacua. Many suspect that these are not real, perhaps because not all moduli are stabilized (see Bena *et al* 2010.10519).

Most of the important problems in statistics of vacua remain open:

- Show that two dual constructions of the same set of vacua produce the same distribution. The examples I know of start from two dual moduli spaces (*e.g.* mirror complex-Kähler) but this should be possible after stabilization.
- Finite number of quasi-realistic vacua (lower bound on KK mass). Good general arguments in Denef/Douglas 0404116 and Acharya/Douglas 0606212, but no proofs.
  Recent progress for flux vacua using powerful math results on CY moduli spaces, see Grimm *et al* 2110.05511.
- Find a way to get a **representative** sample of vacua, in other words the probability distribution for observables approximates that for the full set. Not even clear for $d = 4, \mathcal{N} = 2$. F theory on fourfolds is a good candidate but needs work to be concrete.
- Of course, the basic constructions of quasi-realistic vacua, in particular positive c.c., also have good general arguments but are not yet proven. May need new techniques.

ML for math

Distributions of string EFTs and observables are complicated (maybe impossible) to derive analytically. Can we use ML to learn them? Surely yes, as has been tried in a number of works: He 1706.02714, Carifio *et al* 1707.00655, Mütter *et al* 1811.05993, many more.

There is a related area of generation of mathematical conjectures using ML, which includes the previous works and many more, in particular the recent Nature article of Davies *et al* (the DeepMind group) with results in knot theory and representation theory.

What I want to focus on here, rather than the very interesting physics and math claims from these works, are questions like

- How do we judge and validate or refute such claims?
- How do we reproduce and build on work which uses large computational resources and databases, and depends on potentially fragile code?

ML for math

Distributions of string EFTs and observables are complicated (maybe impossible) to derive analytically. Can we use ML to learn them? Surely yes, as has been tried in a number of works: He 1706.02714, Carifio *et al* 1707.00655, Mütter *et al* 1811.05993, many more.

There is a related area of generation of mathematical conjectures using ML, which includes the previous works and many more, in particular the recent Nature article of Davies *et al* (the DeepMind group) with results in knot theory and representation theory.

What I want to focus on here, rather than the very interesting physics and math claims from these works, are questions like

- How do we judge and validate or refute such claims?
- How do we reproduce and build on work which uses large computational resources and databases, and depends on potentially fragile code?

Judging math/science claims derived using computers

How can we judge a result whose derivation is too long for humans to check? This general question first came to broad attention with the 1976 Appel-Haken proof of the four color theorem.

Now there were many earlier computational results, such as calculations of solutions of ODE's and PDE's, numerical evaluations *etc.*. These were justified by validation on simple (analytically tractable) special cases, and by fitting results into larger patterns with some theoretical understanding. In math, they were considered intermediate and not final results. In physics, there are famous examples of long-standing mistakes, such as $g - 2$ of the muon.

Long calculations and proofs often have mistakes, and thus the Appel-Haken proof was somewhat mistrusted for many years.

Judging math/science claims derived using computers

How can we judge a result whose derivation is too long for humans to check? This general question first came to broad attention with the 1976 Appel-Haken proof of the four color theorem.

Now there were many earlier computational results, such as calculations of solutions of ODE's and PDE's, numerical evaluations *etc.*. These were justified by validation on simple (analytically tractable) special cases, and by fitting results into larger patterns with some theoretical understanding. In math, they were considered intermediate and not final results. In physics, there are famous examples of long-standing mistakes, such as $g-2$ of the muon.

Long calculations and proofs often have mistakes, and thus the Appel-Haken proof was somewhat mistrusted for many years.

There were two main contributors to its final acceptance:

- The proof was independently reproduced and simplified by Robertson *et al* (1997).
- A formal verification was done by Gonthier *et al* (2005).

Formal verification is the expression of a mathematical statement and proof in completely explicit logical terms, such that each step in the reasoning can be verified by a computer. This technology is still not at the point that we could use it for mathematical physics, but this may come in the next few years. See Kevin Buzzard's blog and upcoming ICM lecture.

Judging a claim which involves ML for its formulation involves all of the difficulties of standard computation, plus that of interpretability. ML models are often very difficult to interpret, with millions of parameters and nonlinear interactions. In the simplest case of supervised learning, unless one takes special measures one is interpolating a dataset. Extrapolation of such fits (again without taking special care) is one of the elementary mistakes cautioned against in textbooks.

There were two main contributors to its final acceptance:

- The proof was independently reproduced and simplified by Robertson *et al* (1997).
- A formal verification was done by Gonthier *et al* (2005).

Formal verification is the expression of a mathematical statement and proof in completely explicit logical terms, such that each step in the reasoning can be verified by a computer. This technology is still not at the point that we could use it for mathematical physics, but this may come in the next few years. See Kevin Buzzard's blog and upcoming ICM lecture.

Judging a claim which involves ML for its formulation involves all of the difficulties of standard computation, plus that of interpretability. ML models are often very difficult to interpret, with millions of parameters and nonlinear interactions. In the simplest case of supervised learning, unless one takes special measures one is interpolating a dataset. Extrapolation of such fits (again without taking special care) is one of the elementary mistakes cautioned against in textbooks.

One clear answer to this question would be to use ML only as a source of conjectures, and insist that claims have proofs (according to the standards of the relevant field) before general acceptance.
This criterion is relatively clear for mathematicians, but what about theoretical physicists and other mathematical scientists whose fields do not insist on rigorous proof?

I am not sure what we should do, but here are some ideas:

- Formulate claims in enough generality that they can be tested on exactly solvable examples.
- Don't just test the physically observable predictions, but try to broaden the model to get more testable claims.
- Do not accept uncontrolled extrapolations (in parameters, in dimensions, *etc.*). For example, if one trains on examples with dimension $1 \leq N \leq 20$, to claim that a fit works for $N \gg 20$ one should provide some argument that it has the right asymptotics for large N.

One clear answer to this question would be to use ML only as a source of conjectures, and insist that claims have proofs (according to the standards of the relevant field) before general acceptance.
This criterion is relatively clear for mathematicians, but what about theoretical physicists and other mathematical scientists whose fields do not insist on rigorous proof?

I am not sure what we should do, but here are some ideas:

- Formulate claims in enough generality that they can be tested on exactly solvable examples.
- Don't just test the physically observable predictions, but try to broaden the model to get more testable claims.
- Do not accept uncontrolled extrapolations (in parameters, in dimensions, *etc.*). For example, if one trains on examples with dimension $1 \leq N \leq 20$, to claim that a fit works for $N \gg 20$ one should provide some argument that it has the right asymptotics for large N.

Reproducibility and extensibility

Ultimately, the value of a scientific result is entirely dependent on the ability of others to reproduce and extend it. As ML plays a larger role in the mathematical sciences, and as its results become more interesting, this simple point will become more and more important.

ML already has problems in this regard. Some of the most interesting results, such as training of language models, require computational resources which are beyond the reach of almost every academic group. We might add to this the need for engineering expertise which is only available in leading industrial laboratories.

Long before the development of artificial general intelligence (in 30 years?), we can imagine disturbing possibilities. Suppose an industrial lab (DeepMind?) uses computer assistance to find a proof of the Riemann hypothesis. What impact would this have on the academic world?

Reproducibility and extensibility

Ultimately, the value of a scientific result is entirely dependent on the ability of others to reproduce and extend it. As ML plays a larger role in the mathematical sciences, and as its results become more interesting, this simple point will become more and more important.

ML already has problems in this regard. Some of the most interesting results, such as training of language models, require computational resources which are beyond the reach of almost every academic group. We might add to this the need for engineering expertise which is only available in leading industrial laboratories.

Long before the development of artificial general intelligence (in 30 years?), we can imagine disturbing possibilities. Suppose an industrial lab (DeepMind?) uses computer assistance to find a proof of the Riemann hypothesis. What impact would this have on the academic world?

Reproducibility and extensibility

Ultimately, the value of a scientific result is entirely dependent on the ability of others to reproduce and extend it. As ML plays a larger role in the mathematical sciences, and as its results become more interesting, this simple point will become more and more important.

ML already has problems in this regard. Some of the most interesting results, such as training of language models, require computational resources which are beyond the reach of almost every academic group. We might add to this the need for engineering expertise which is only available in leading industrial laboratories.

Long before the development of artificial general intelligence (in 30 years?), we can imagine disturbing possibilities. Suppose an industrial lab (DeepMind?) uses computer assistance to find a proof of the Riemann hypothesis. What impact would this have on the academic world?

Without wandering off into science fiction, we can still foresee that we have unprecedented opportunities to enhance our capabilities for the discovery and propagation of knowledge, if we properly judge the situation and focus our attention on developments which will not quickly be made obsolete by the advance of technology, but rather will play more central roles in the future.

Although much of the value of ML/AI comes from the ability of these systems to examine huge datasets and find new patterns, a significant part just comes from reproducibility and extensibility, the ability to upload a working model (say to Github) and watch as the rest of the world adopts it, adapts it and improves it, without the need to re-implement it or even to deeply understand the thinking that created it.

The rest of the mathematical sciences have some catching-up to do in this regard. How much time does it take you to learn and use the innovations in your colleagues' string theory papers? And how much time to learn a new ML package?

Without wandering off into science fiction, we can still foresee that we have unprecedented opportunities to enhance our capabilities for the discovery and propagation of knowledge, if we properly judge the situation and focus our attention on developments which will not quickly be made obsolete by the advance of technology, but rather will play more central roles in the future.

Although much of the value of ML/AI comes from the ability of these systems to examine huge datasets and find new patterns, a significant part just comes from reproducibility and extensibility, the ability to upload a working model (say to Github) and watch as the rest of the world adopts it, adapts it and improves it, without the need to re-implement it or even to deeply understand the thinking that created it.
The rest of the mathematical sciences have some catching-up to do in this regard. How much time does it take you to learn and use the innovations in your colleagues' string theory papers? And how much time to learn a new ML package?

So the answer to the question “What is string data?” is, whatever helps us to most quickly make discoveries, build on each other’s discoveries, and somehow get a grip on what at present seems like a vast problem.

The core of this is to validate and save the datasets we generate in our research, but I think there will be just as much payoff to better automating the parts of the problem we already understand – basic computations in geometry and group theory, perturbative calculations for QFT, interactive proofs of those results which we can prove. This is not to say that we understand everything well, but rather that we understand some essential parts well enough to do this.

Looking ahead 10 years, while I am not sure whether we will have evidence for or against string theory, I am pretty sure that we will be reading our papers in a different way – perhaps downloading the associated notebook first, or perhaps letting the computer summarize just the parts which are new to each of us. As they were with so many other developments, physicists can be in the forefront here as well.