

Mathematics - a new Domain for Datamining?

Simon Colton
Division of Informatics
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
Scotland

Abstract

With the many databases of mathematical information currently available, there is much potential for datamining techniques to find new and interesting mathematical results. Indeed, we suggest that, if we can utilise the research into dealing with dynamic, distributed and heterogeneous datasets, datamining could be as successful a technique for mathematics as it is for, say, biology. We briefly survey 7 mathematical databases available online and present a motivating example and a case study. This enables us to highlight important issues and to make some suggestions for datamining mathematical information.

Introduction

There have been many calls to formalise mathematics in order to have an all encompassing database of mathematical knowledge. For example, the mission statement taken from the QED project web site¹ states:

‘The aim ... is to build a single, distributed, computerized repository that rigorously represents all important, established mathematical knowledge.’

The benefits of such a database for automated theorem proving and computer algebra would be great — clearly such a knowledge base could enhance any mathematics program. Given such a knowledge base, it may also be possible to induce new and interesting conjectures using datamining techniques.

Unfortunately, no such database exists, and while there are still projects to formalise mathematics, the projects now have smaller ambitions. One problem, of course, is the sheer amount of mathematical data available: in (Hoffman 1999) it is estimated that around 250,000 new theorems are proved and presented in journals every year. Another stumbling block has been the choice of language/logic in which to formalise the concepts and theorems. For instance, the QED project, for which there was much initial interest, has largely ended because a logic could not be agreed upon.

Copyright © 2001, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

¹Found here: <http://www-unix.mcs.anl.gov/qed/>

However, various ad-hoc mathematics databases have been constructed, which we survey below. We believe that it is now possible to think of datamining this wealth of knowledge to find interesting new conjectures. Some new mathematical databases are being proposed and developed, and we hope that our proposals can also be used to shape their development.

A Motivating Example

To indicate how, in theory, very simple datamining techniques could lead to new results, we can take motivation from one of the highlights of mathematical research in recent years. In 1998, Richard Borcherds received the most prestigious prize in mathematics, the Fields medal. This was awarded for his work proving the ‘Monstrous Moonshine’ conjectures originally discovered by John McKay and developed, by, amongst others, John Conway (Conway & Norton 1979).

What is remarkable about the Moonshine conjectures is that they came about by accident: John McKay noticed that a coefficient in the j -function (an elliptic modular function) was one more than a number in the characteristic table of the Monster group in group theory. This wouldn’t have been so remarkable if the numbers hadn’t been so large: 196,883 and 196,884. McKay wondered whether this was a mere coincidence, largely because group theory and modular functions were so far removed as domains of pure mathematics. However, he, John Conway and others began to realise that it was not a coincidence, but represented a deep connection between these two previously disparate domains. As an aside, it is quite fitting that Borcherd’s eventual proof of the connection also combined notions from very varied sources, in particular string theory from physics.

It is easy to see how this conjecture could have been found automatically if there were mathematical databases of elliptic functions and characteristic tables available for datamining. A simple exhaustive search looking for large numbers common to both databases, (within a small margin of error), would have found this conjecture. This very much simplifies the combinatorial complexities which might arise in doing such an exhaustive search. However, the search could be done in parallel, or even distributed over the internet, like

the various projects² to find large prime numbers. It is likely that there would be at least as much interest in finding a new and interesting conjecture as there is in finding a new prime number. We argue below that the problem is probably not computing power, but rather the nature and extent of the mathematical databases available and the inability of computers to identify the more interesting results.

Survey of Mathematical Databases

There are many different types of mathematical information, including (i) objects, such as integers, groups or geometrical figures (ii) concepts, such as prime numbers, Abelian groups, equilateral triangles (iii) conjectures, such as: cyclic groups are Abelian (iv) proofs to theorems and (v) algorithms. The largest volume of knowledge resides in the mathematical journals, which contain papers with all the different types of information in them. Some papers are available on the internet, but as postscript files, they are mainly intractable.

Fortunately, there are also various databases containing large collections of information in particular domains of mathematics. We briefly survey 7 such databases below which have online access, in order to give an impression of how distributed, dynamic and heterogeneous the online knowledge base of mathematical knowledge is. They are ordered roughly in increasing terms of the generality of the information they contain.

- Eric Weisstein's Mathworld

Mathworld (<http://mathworld.wolfram.com>) is an online mathematics Encyclopedia is a very valuable source of information. When it was last reviewed by the NetSurfer magazine, it contained 8,974 entries along with 153,958 cross-references, 3,639 figures, and 917 Java applets. The 1400 pages are based around concepts, with definitions, examples, algorithms, related theorems and references given for each concept. The Encyclopedia was published in book form in 1998, and unfortunately, due to a copyright dispute with CRC publishers, the web site is presently not online.

- MathSciNet

The American Mathematical Society maintain the MathSciNet citation and review server at <http://www.ams.org/mathscinet>. At present, it contains reviews of 10,843 mathematics articles with references for 151,350 articles in total, written by 358,104 authors in 1,799 different journals.

- The Mizar Library

The Mizar Project has been underway since 1973 and aims to formalise and verify mathematics articles in terms of Tarski-Grothendieck set theory (see <http://mizar.org/library/>). Since 1989, formalised

articles have been collected and at present, there are 666 articles, contributed by 120 people. The articles contain over 2000 different concept definitions and each article is written up as a paper in the Journal of Formalised Mathematics. Each article references others and often repeated references are made. The articles are then ranked on the web site in terms of the number of references they have, which gives some indication of how important the article is. For instance, the top article (not surprisingly, the axioms of set theory) is referenced 5813 times in the database, with the 50th article referenced only 751 times (as of March 23rd 2000). Another useful piece of information is a net of structures showing which structures are built from which others.

- The Mathematica Library

Mathematica is a computer algebra system (CAS) which comes with over two thousand algorithms as standard (see <http://www.wolfram.com>). The algorithms are optimised for performing complex calculations including algebraic manipulation and many very specialised functions. While many algorithms are hard-coded into Mathematica, many others are stored in the Mathematica language which can be accessed through the Mathematica program or the MathReader (a free download). In addition, there are hundreds of worked examples and tutorials available for download. There are libraries of functions for all the major computer algebra systems, and many of these libraries are available as ASCII text documents which can be loaded into the CAS.

- The Encyclopedia of Integer Sequences

This database, maintained at <http://www.research.att.com/njas/sequences/>, contains over 60,000 integer sequences (such as prime numbers, square numbers, etc). They have been collected over 35 years by Neil Sloane, with contributions from hundreds of mathematicians. The Encyclopedia is very popular, receiving over 16,000 queries every day. The first terms of each sequence are stored, and this is how the database is queried: the user provides the first terms of a sequence they wish to know more about. In addition to the terms of the sequence, a definition is also given and keywords such as 'nice' (intrinsically interesting) and 'core' (fundamental to number theory or some other domain). There are also links to other sequences under a "see also" heading, and some sequences are supplied with code for one of a number of computer algebra systems, such as Mathematica. The database is fairly heterogeneous, as the number of terms for each sequence differs from 2 or 3 to 200 or 300, some sequences have program code, but most don't and each contributor has his or her own style for the definitions.

²See the Great Internet Mersenne Prime Search at <http://www.mersenne.org/prime.htm>, for instance.

- The Inverse Symbolic Calculator

Given a constant (decimal number), this calculator attempts to find a match to one in its database (<http://www.cecm.sfu.ca/projects/ISC/>). It uses a set of 400 tables of constants and a set of programs for transforming the given constant in order to find a match. The tables hold more than 50 million constants taken largely, but not exclusively, from mathematics and the physical sciences. Each constant is stored alongside a name, a definition, references, and where possible, a program to generate the digits of the constant. The programs (known as the ‘smart lookup’) perform around 200 transformations.

- The Geometry Junkyard

This site contains research papers, abstracts, programs, problem statements, lecture notes and web pointers to all things to do with geometry. The pages are sorted into 29 subheadings and each one has a number of pages attached. There is an abundance of web sites such as this which contain mathematical tutorials, exercises, FAQ sheets and open questions. They are usually restricted to a particular domain such as geometry, group theory, number theory, etc., but sometimes they cover larger areas of mathematics.

Case study: Datamining the Encyclopedia of Integer Sequences

Automated reasoning techniques and search algorithms have contributed to mathematics by discovering proofs to well known theorems and solving existence problems [see (McCune 1997) and (Slaney, Fujita, & Stickel 1995) respectively]. In contrast, very little research has been undertaken towards using datamining techniques for knowledge discovery in mathematical databases. To our knowledge, the only such attempt has been our work datamining conjectures from the Encyclopedia of Integer Sequences, as discussed in (Colton 1999) and (Colton, Bundy, & Walsh 2000a).

Given a sequence of integers, S — usually one invented by our HR program (Colton, Bundy, & Walsh 1999) — we searched through the Encyclopedia to find others which were (i) subsequences of S (ii) supersequences of S and (iii) disjoint with S . For example, looking for supersequences of the perfect numbers³ (not one invented by HR), we found that sequence A009242 is a supersequence. This is the sequence obtained by taking $n = 1, 2, 3, \dots$ and writing the lowest common multiple (lcm) of n and $\sigma(n)$, where $\sigma(n)$ is the sum of the divisors of n . Hence the datamining had produced the following conjecture: all perfect numbers can be written as $lcm(n, \sigma(n))$ for some n . We proved this result and published it, along with others produced by datamining, in (Colton 1999).

³Integers such as 28, which equal the sum of their proper divisors ($28 = 1+2+4+7+14$).

We found that the main problem with this approach was not generating conjectures, but pruning those which were uninteresting. For example, looking for supersequences of the perfect numbers as above produces 7710 examples (in 927 seconds on a 500Mhz Pentium processor). The details of the measures for pruning are given in (Colton, Bundy, & Walsh 2000a). It suffices here to remark that we used:

- The terms of the sequence: for instance, we often specified that there must be at least three terms in the supersequence which are perfect numbers.
- The definition of the sequence: for instance, we discarded any supersequences if the definition contained the word “perfect” because the conjecture arising would probably be uninteresting.
- Values of judgement, namely the keywords associated with each sequence. For instance, we restricted the output to only those with the ‘nice’ or ‘core’ keywords in their Encyclopedia entry.
- Contextual information. As discussed above, each sequence is stored in the Encyclopedia along with a list of other (“see also”) sequences. We used this information in two ways: (i) to discard any supersequences which are linked to perfect numbers, as the resulting conjecture will probably be obvious and (ii) to discard any supersequences which are not linked to any other sequences, because they will probably not be interesting.

We found the measures based on the terms of the sequence and those based on the keywords to be most effective in pruning the output. For instance, specifying that at least three terms of the supersequence be perfect numbers reduced the yield from 7710 to just 47 (including the *pernicious* numbers — where the binary representation contains a prime number of 1s — which eventually led us to prove the appealing conjecture that perfect numbers are actually pernicious). However, the pruning measures based on words in the definition and the contextual information also came in useful at various stages of our investigation.

Issues in Datamining Mathematical Information

To make general comments about datamining mathematics databases, we imagine a set of databases containing (i) examples of mathematical objects, e.g. integers (ii) concepts about those objects, e.g. the concept of square numbers (iii) conjectures e.g. square numbers have an odd number of divisors, (iv) proofs to theorems, e.g. the proof that square numbers have an odd number of divisors and (v) algorithms, e.g. the sieve of Eratosthenes for generating prime numbers. Furthermore, each database contains information from one domain only. This last restriction simplifies matters, but, given the cross domain nature of mathematics, it will be hard to uphold in practice. The kind of con-

jectures we would hope to mine from these databases are of the form: “Concept X in Domain D is related to Concept Y in Domain E, because of Z”. For instance, the Moonshine conjecture would be: “the j -function in modular functions is related to the character table in group theory because of the number 196,833 ...”

As we saw when datamining the Encyclopedia of Integer Sequences, a notion of what is and what isn’t an interesting conjecture is vital to prune the output. In (Colton, Bundy, & Walsh 2000b) we suggest some general ways in which the interestingness of a conjecture could be estimated. Four such measures are:

- Complexity of examples. If the numbers in the Moonshine example had been 17 and 18 rather than 196,884 and 196,833, it is much more likely to have been just a coincidence than an indication of a deep connection. Hence a measure of interestingness of a conjecture could be based on the complexity of the examples which led to the conjecture. This would suggest storing examples in order of decreasing complexity, so the most complicated examples are used to find conjectures before the less complicated ones.

- Complexity of the conjecture statement. We can also consider the complexity of the conjecture statement. However, here the reverse is usually true: often the most simply stated conjectures, such as Goldbach’s conjecture or Fermat’s last theorem, are the most interesting. This would suggest storing concepts in the database in increasing order of complexity (as conjectures about simple concepts are usually simpler than conjectures about complicated concepts).

- Surprisingness across domains. Surprising conjectures are often the most interesting⁴ ones. A conjecture might be surprising if (like the Moonshine conjectures) it suggests a relationship between two very different domains of mathematics. This would suggest maintaining meta-information about the databases, such as how related each database is to the others.

- Surprisingness within a domain. A conjecture might also be surprising if it related two concepts in very different areas of the same domain. This suggest storing some kind of semantic net within a database, similar to how sequences can be linked in the Encyclopedia of Integer Sequences and to how the Graffiti program discards trivial conjectures with it’s Dalmation heuristic (Fajtlowicz 1988).

It is important to note that the Moonshine conjectures arose not by involved study of the definitions of the concepts in both domains, but rather by finding a similarity in the examples of the concepts. Drawing from this example, we note that in every domain of mathematics, coefficients are calculated and objects are described with numbers. This would suggest a starting

⁴John Conway is quoted in (Fajtlowicz 1999) as saying that the best conjectures are “outrageous”.

point for datamining across datasets would be to look for common numbers. This would need some statistical margin of error, so that, for instance, the commonality between 196,884 and 196,833 is found. The allowed error could be relative to the size of the numbers, so that very large numbers differing by, say, only 10 are identified. Groups are also ubiquitous in pure mathematics, found in domains ranging from Relativity to Galois theory. Hence, it could be instructive to find commonalities between groups in different datasets. Margin of error here is less obvious, but it could be in terms of the size of the group and/or the family to which it belongs (e.g. symmetric, cyclic, dihedral, etc.)

Another important issue is how to cross reference information between databases of different information types, for instance a database of formalised mathematics and a database of integer sequences. In addition to the usual problems of noise, incorrect and missing data common to any large knowledge base, there are also an overwhelming number of inconsistencies in the mathematical literature⁵, including (a) different names for the same concept/object/conjecture (b) different algorithms for the same calculation (c) different representations of the same objects and (d) different formats for the same conjecture. Hence cross referencing will be a difficult process and it is possible that bespoke translators will have to be set up between pairs of databases.

Much could be done to improve mathematics databases if certain considerations were taken into account in their construction. In particular, sticking to some common format for concept definitions and conjecture statement would make datamining much easier. This does not necessarily have to involve complete formalisation in a particular logic, rather it involves sensible database administration. For instance, there are many different formats for conjectures, but it may be possible to write many conjectures in just three formats:

- Concept A if and only if Concept B
- Concept A implies Concept B
- The only examples of Concept A are $\{a_1, \dots, a_n\}$ (where n could be zero).

The majority of theorems found in mathematics texts can be interpreted in one of the three ways. Making explicit which concepts are involved in the conjectures in this manner would improve the chances of cross-referencing data. Another improvement would be to add extra information to the data already available. For instance, adding details of which concepts are being calculated by an algorithm would make cross-referencing easier.

⁵Which is a motivation for formalisation efforts.

Conclusions and Future Work

We have surveyed some mathematical databases and given a brief overview of our attempts to datamine one of them. This has enabled us to identify some initial pointers for datamining sources of mathematical knowledge:

- Automated discovery in mathematics does not necessarily have to be based on the formal definitions of concepts and statements of conjectures. Effective techniques using the examples of concepts could be implemented. Given the general unwillingness to agree on a formalism for mathematical databases, using examples may be the only viable approach to datamining.
- An intelligent database would need a strong aesthetic, i.e. be able to extract the interesting conjectures from a plethora of dull results. This will require measures of interestingness based on examples, definitions, user-supplied values of worth and other contextual information, such as a semantic net linking concepts.
- Looking for common numbers and possibly groups across data sets would be a good starting point. Some margin of error would be needed.
- In addition to examples and formal statements of concepts and conjectures, databases should (at least) contain semantic information conveying context and the relative worth of concepts and conjectures.
- Some standardisation of the knowledge will be required in order to effectively data mine the databases. This will not necessarily require a complete formalisation, but rather writing conjectures and concepts in similar formats, agreeing on a single name for a concept/conjecture, providing additional information and so on.
- It may be a good idea to distribute searches for conjectures over the internet, as there is much interest in large searches for mathematical novelties.

Mathematical databases have the same problems as other scientific datasets: they are heterogeneous both internally and in relation to each other. They are also dynamic: for instance, the Encyclopedia of Integer Sequences often gains scores of new sequences in a single day, and MathSciNet has added 19,878 items in the year 2001 alone. They are also distributed at various web sites worldwide. We have not added to the discussion of how to deal with such data in general. Rather, we have proposed mathematics as a domain for study of knowledge discovery. To emphasise our proposal, we gave a motivating example which would require datamining over distributed databases, and we showed that datamining the Encyclopedia of Integer Sequences has resulted in conjectures worthy of the mathematical literature.

We hope to learn from the research into distributed, dynamic and heterogeneous knowledge sources being

undertaken in Artificial Intelligence. One of many applications of this research would be to add value to mathematical databases, by enabling the database to make conjectures whenever a new data item was inserted. We see no reason why an intelligent database of, say, elliptic functions, when given the coefficients of the j -function shouldn't remark:

'I noticed in the group theory database that coefficient 196,884 is one more than 196,883, a number which appears in the characteristic table of the Monster group. Is this interesting?'

Such a system is a long way off, but we hope to have provided some directions to follow in order to apply datamining to mathematical databases.

Acknowledgments

The author is also affiliated to the Department of Computer Science at the University of York. This research is supported by EPSRC grant GR/M98012.

References

- Colton, S.; Bundy, A.; and Walsh, T. 1999. HR: Automatic concept formation in pure mathematics. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*.
- Colton, S.; Bundy, A.; and Walsh, T. 2000a. Automatic invention of integer sequences. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*.
- Colton, S.; Bundy, A.; and Walsh, T. 2000b. On the notion of interestingness in automated mathematical discovery. *International Journal of Human Computer Studies* 53(3):351–375.
- Colton, S. 1999. Refactorable numbers - a machine invention. *Journal of Integer Sequences* 2.
- Conway, J., and Norton, S. 1979. Monstrous moonshine. *Bulletin of the London Mathematical Society* 11:308 – 339.
- Fajtlowicz, S. 1988. On conjectures of Graffiti. *Discrete Mathematics* 72 23:113–118.
- Fajtlowicz, S. 1999. The writing on the wall. Unpublished preprint, available from <http://math.uh.edu/clarson/>.
- Hoffman, P. 1999. *The man who loved only numbers*. Fourth Estate.
- McCune, W. 1997. Solution of the Robbins problem. *Journal of Automated Reasoning* 19(3):263–276.
- Slaney, J.; Fujita, M.; and Stickel, M. 1995. Automated reasoning and exhaustive search: Quasigroup existence problems. *Computers and Mathematics with Applications* 29:115–132.