DANGER
(and thoughts on [MPP21])

Notes by J.H.Davenport

25–26 August 2021

Contents

The workshop details are at `https://sites.google.com/view/danger-workshop/talks?authuser=0`. JHD only attended live for day 1 (August 25th).

Chapter 1

August 25, 2021

1.1 Minhyong Kim (University of Warwick), How hard is it to learn a mathematical structure?

Slides at `https://drive.google.com/file/d/13miyHguZSHn9NMTwPi3n7zFYtFsfSQhy/view?usp=sharing`; video at `https://www.youtube.com/watch?v=uHeIFxR6inM`.

See paper with He, 1905.02263 [HK19]. Look initially at Finite Groups. A finite group is an *associative* normalised Latin Square. Note that associativity is tedious in practice. But a machine seems to do small ones easily.

Other challenges: recognise manifold from weak manifold. More generally, given a first-order theory T, can we recognise a model of T? Does this differ between stable T and ustable T? (He didn't define stable).

1.2 Henry Adams (Colorado State University), Applied topology: from global to local

Slides at `https://www.math.colostate.edu/~adams/talks/AppliedTopology-FromGlobalToLocal2.pdf&sa=D&sntz=1&usg=AFQjCNExkFzO9a2Py_4qKD_s6qZIUo9GOw`; video at `https://www.youtube.com/watch?v=ZUU41-kTHyc`.

Flock of, say, 1M birds, so $\begin{pmatrix} N \\ 2 \end{pmatrix}$ distances, etc. To visualise the flock, look at persistent homology. "A geometer can tell the difference between a coffee mug and a doughnut" — areas of negative curvature, etc. H_0 counts components, H_1 counts loops, H_2 counts voids etc. Persistent homology will add edges if things are "near", and as this changes we get a change in homology: features like loops are born then die as "near" grows. The PH algorithms are cubic, but this can be expensive. Can ML help?

Example of H_0 and H_1 for brain arteries across age.

Most of my work has been on the fairly explainable end of ML, e.g. k-means. Feeding a persistence diagram into ML is harder than one thinks. One has birth/death events, but how to encode? ML learns from the short bars. There's a theorem [Bub15] that persistent homology can detect curvature. Also [Adamsetal2020a] can detect fractal dimension.

Example: Aphids on leaves, and able to reject hypothesis of independence.

1.3 Sonja Petrović (Illinois Institute of Technology), Learning in commutative algebra & models for random algebraic structures.

Video at `https://www.youtube.com/watch?v=kz9TQeSZt2Q`.

Useful pre-lecture chat on teaching in a post-Covid world.

One question is whether I can use statistics/ML to improve Buchberger's algorithm. Two-step process: *learning* on algebraic structures over *interesting* distributions. Long history of randomness, e.g. [LO38]. Schwartz–Zippel Lemma. See also [PSHL20]. Strongly suggest [Pei21] as a starting point. Formulation of Buchberger as RL was amazing. A key part of RL is predicting future reward. Our work is in [MPP21].

Problem 1 *Given I, how many additions within one run of Buchberger. JHD notes that all this work is in the context of binomial ideals, so all polynomials stay binomial, and #additions* is *a reasonable measure of complexity. This wouldn't be the case for non-binomial ideals, where the lengths of the polynomials would also have to be taken into account.*

Use MMMS (min/max/mean/s.d.) of degrees as criterion. Even for linear regression as our modelling technique, this has better R^2 than "regularity", and ML does better.

We needed a formal probabilistic model for our random structure. Note Erdős–Renyi–Gilbert random graph model $G(n,p)$ [n=#vertices, p=probabiity of an edge]. Similarly, can have a model for random monomial ideals [DLPS$^+$19]. Then can get probability of a given ideal. [DLPS$^+$19] also gives similar on simplicial complexes, and for a suitable choice recovers the Costa–Farber model [CF16], for example. Krull dimension has a threshold behaviour: $D^{-t} \ll p << D^{-t+1} \Rightarrow \dim(S/I) = t$ asymptotically always surely. Similar behaviour for Betti numbers (understood). Also (not so well understood) threshold behaviours for projective dimension, %age Cohen–Macaulay (30%?).

1.3.1 Q&A

Q–ME How important was it that you had random data?

A If you want performance on random problems, yes.

Q–JHD Is this the right probability model, e.g. real roots in one variable?

A Certainly not, but it's a start.

1.4 Siu-Cheong Lau (Boston University), Deep learning over the moduli space of quiver representations

Slides at https://drive.google.com/file/d/1dZfnP8oCwe_YNUnGcwlqq_Rfp6SY3hwB/view?usp=sharing; video at https://www.youtube.com/watch?v=QU5KhpPYk6I.

Note that a NN looks rather like a directed graph, and this is a quiver representation (vertices are vector spaces and arrows are linear maps). Fixing such a map for each arrow, we can get ...

Given $K \subset V_{in}$ and a continuous $f : K \to V_{out}$, want to minimise some cost. Note that composition of non-linear functions is not distributive, hence we only have a near-ring. $R(Q)$ is the space of representations, but generlaly work in the moduli space, $M(Q) = \{\text{stable points in } R(Q)\}/G$ where $G = \prod_i GL(V_i)$. Can we actually do DL here?

But the cost function isn't necessarily well-defined on $M(Q)$ rather than $R(Q)$. See 2101.11487 for a solution.

Framed representations have linear maps $e_i : \mathbf{C}^{n_i} \to V_i$. This gives us R^{rf} and corresponding M^{rf}.

Speaker dropped at this point, and JHD had to leave.

Chapter 2

26 August

2.1 Riccardo Finotello: Algebraic geometry and computer vision: inception neural network for Calabi-Yau manifolds

Slides at https://drive.google.com/file/d/1B0-lPBQI5gQhZRyPvM1YH_VRD9QioPtm/view; video at https://www.youtube.com/watch?v=m9b3HOUjRJU.

2.2 Kyu-Hwan Lee: Applications of machine learning to data from number theory

Slides at https://drive.google.com/file/d/1nPlqxBVASylav7-Rq1GZ_xUUZ2Sf7v7j/view?usp=sharing; video at https://www.youtube.com/watch?v=Wa4g2ulxkQM.

Based on [HLO20c, HLO20b, HLO20a].

2.3 Roozbeh Yousefzadeh: Deep learning generalization, extrapolation, over-parameterization and decision boundaries

Video at https://www.youtube.com/watch?v=Wa4g2ulxkQM.

2.4 Ruriko Yoshida: Tree topologies along a tropical line segment

Slides at https://drive.google.com/file/d/1DOsl2SfaPkYtYq5DQR3yIiI61drNwZPy/view?usp=sharing; video at https://www.youtube.com/watch?v=Xmy9isYZCDU.

Chapter 3

JHD's thoughts on [MPP21]

MMMSD is [min,max,mean,standard deviation] of the degree of the generators, used as a simple predictor for linear regression.

1. As stated in Problem 1, this (and [PSHL20]) are based on binomial ideals. Hence the polynomials themseles do not grow, and #additions is a reasonable metric. For non-binomial ideals, one should really consider length of the polynomials, even though no current heuristics do directly (though the tie-breaking technique for the "normal" strategy [GMN+91] is to choose the oldest polynomials, with the argument that these probably have least fill-in).

2. For me, the real conclusions were in Table 7 (using MMMSD) and Table 9 (using neural networks).

3. Table 7 for binomial ideals seems to show that there is almost no point for unweighted degree sampling. I suspect this is because in unweighted sampling there's almost always a generator of maximal degree (see Figure 2c) and in general there's not that much variation.

4. Conversely MMMSD seems to have pretty good results for the weighted distributions *when tested on the same distribution.* This is not altogether surprising: one would expect edgree to predict running time quite when *when* there's a range of degrees.

5. For the toric ideals in table 7, again the diagonal entries are pretty respectable. The bottom-right corner is odd, though. The model trained on $T(6, 0, 10, 8)$ performs 0.88 when tested on the same, and 0.52 when tested on $T(6, 0, 5, 8)$. This is not much worse than the model trained on $T(6, 0, 5, 8)$, which scores 0.64 against itself. *But* that model scores -0.61 when tested against $T(6, 0, 10, 8)$: an amazing lack of symmetry.

6. When it comes to the neural nets (Table 9), the binomial ideals do better than MMMSD when trained and tested on themselves. This time 3-20-10-w performs positively when tested on 3-20-10-u, and *vice versa*, and we see the same behaviour for 3-20-4-w/u.

7. On the toric ideals, the situation is odd, even for the diagonal entries. Both $T(4, 0, 5, 8)$ and $T(6, 0, 10, 8)$ have neural nets that do significantly *worse* than MMMSD-based linear regression.

8. The toric off-diagonal entries are distinctly unimpressive.

9. The base conclusion seems to be that there is little portability of models between data sets, which is depressing. I wonder what would have happened had a model been trained on multiple data sets.

10. Unlike [PSHL20], there is no general principle deduced. I would like to understand how that was deduced in this case.

Bibliography

[Bub15] P. Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16:77–102, 2015.

[CF16] A. Costa and M. Farber. Large random simplicial complexes, I. *Journal of Topology and Analysis*, 8:399–429, 2016.

[DLPS+19] J.A. De Loera, S. Petrović, L. Silverstein, D. Stasi, and D. Wilburne. Random monomial ideals. *Journal of Algebra*, 519:440–473, 2019.

[GMN+91] A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso. One sugar cube, please, or selection strategies in the Buchberger algorithm. In S.M. Watt, editor, *Proceedings ISSAC 1991*, pages 49–54, 1991.

[HK19] Y.-H. He and M. Kim. Learning Algebraic Structures: Preliminary Investigations. `https://arxiv.org/abs/1905.02263`, 2019.

[HLO20a] Y.-H. He, K.-H. Lee, and T. Oliver. Machine-learning arithmetic curves. `https://arxiv.org/abs/2012.04084`, 2020.

[HLO20b] Y.-H. He, K.-H. Lee, and T. Oliver. Machine-learning number fields. `https://arxiv.org/abs/2011.08958`, 2020.

[HLO20c] Y.-H. He, K.-H. Lee, and T. Oliver. Machine-learning the Sato–Tate conjecture. `https://arxiv.org/abs/2010.01213`, 2020.

[LO38] J. Littlewood and A. Offord. On the number of real roots of a random algebraic equation. *J. London Math. Soc.*, 13:288–295, 1938.

[MPP21] J. Mojsilović, D. Peifer, and S. Petrović. Learning a performance metric of Buchberger's algorithm. `https://arxiv.org/abs/2106.03676`, 2021.

[Pei21] D. Peifer. *Reinforcement Learning in Buchberger's Algorithm.* PhD thesis, Cornell University, 2021.

[PSHL20] Dylan Peifer, Michael Stillman, and Daniel Halpern-Leistner. Learning selection strategies in buchberger's algorithm. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7575–7585. PMLR, 13–18 Jul 2020. URL: https://proceedings.mlr.press/v119/peifer20a.html.