Miguel O'Malley

Research Statement

In my research, I focus on emergent methods with impact in machine and human learning, as well as prospective methods in data analysis. Towards more efficient computation on both ends, I focus on topological methods which provide access to new and interesting information for machine learning, such as shape, and mathematical reasoning for humans, through projects such as the fence challenge.

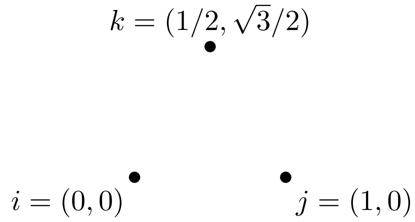
For the purpose of machine learning and data analysis, I investigate properties of metric spaces through the lens of topological data analysis and magnitude. An isometric invariant of metric spaces, magnitude has been shown to encode a number of other valuable invariants, such as dimension and curvature. In particular, magnitude is known to be strongly connected to Minkowski dimension for positive definite compact metric spaces. It stands to reason that magnitude could be leveraged to estimate the dimensions of compact metric spaces from which point clouds are sampled. However, the computational complexity of magnitude renders this prospect nearly impossible to realize for metric spaces of larger size. In recent work with Sara Kalisnik and Nina Otter, we identify alpha magnitude, an invariant arising from topological data analysis and inspired by magnitude, as a method which provides a potential solution to this problem at substantially reduced computational complexity.[OKO22]

Another topic I have worked on involves the application of reinforcement learning, evolutionary algorithms, and A* search for the purpose of solving a higher dimensional sliding puzzle. Multi-agent pathfinding is known to be NP-hard in even ideal cases, but to add further to the complexity, the cubical sliding puzzle defeats existing complete methods through changing the underlying space of valid moves based on the positions of marked vertices. Our implementations of RL and EA provide an implementation of ML-based MAPF in exceedingly diverse environments, implying myriad possible future directions for the implementation of both in learnable, NP-hard problems.[MOMR24]

Finally, for the purposes of outreach and measuring human learning, we have produced a full-stack web development game project, which we have termed the Fence Challenge. The pentomino, a polyominoid with 5 squares (the domino has 2) has 12 different configurations, up to rotation and reflection. The question of the maximal area which can be enclosed by the 12 different pentominos is a version of the isometric problem, for which the answer is known. However, the answer is not known for all orders the pentominos can be placed in, for smaller subcollections of the pentominos, nor is the order of pentominos producing the lowest maximal area known. These are all questions we seek to understand human performance on through our implementation of the fence challenge as a game, playable online or in person.

Alpha Magnitude

Introduced in 2011 by Tom Leinster [Lei13], magnitude is an isometric invariant of metric spaces. The computation for finite spaces is fairly straightforward. For metric space X, we begin with a distance matrix D where each entry $d_{i,j}$ represents the distance between point i and point j in space X.



A space of 3 equidistant points...

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

...and its distance matrix.

$$\begin{bmatrix} 1 & e^{-1} & e^{-1} \\ e^{-1} & 1 & e^{-1} \\ e^{-1} & e^{-1} & 1 \end{bmatrix}$$

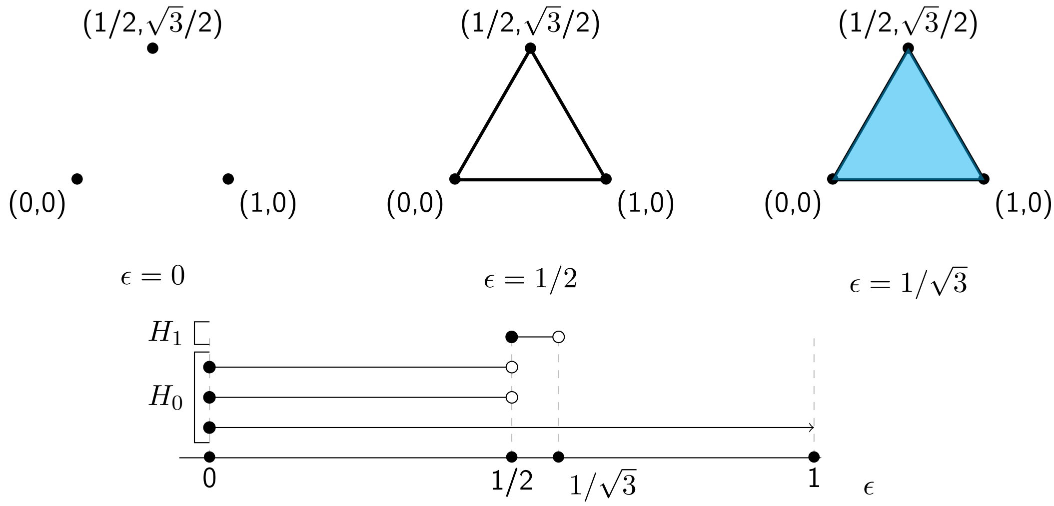
The similarity matrix.

To compute magnitude we first take the negative exponential of each entry of the above distance matrix, referred to as a similarity matrix. We then find a vector such that the product of the vector and the above matrix has only 1 in each entry. We refer to this vector as a weighting.

$$\begin{bmatrix} 1 & e^{-1} & e^{-1} \\ e^{-1} & 1 & e^{-1} \\ e^{-1} & e^{-1} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{1+2e^{-1}} \\ \frac{1}{1+2e^{-1}} \\ \frac{1}{1+2e^{-1}} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

A weighting for the above metric space.

The sum of the entries in the weighting is referred to as the magnitude. Higher magnitudes correspond to more less clustered spaces, and vice versa. But this computation required us to solve an $Ax = b$ matrix equation. Magnitude is suitable for computation in spaces of lower cardinality, but as spaces become more populous magnitude becomes substantially more expensive, rendering it computationally infeasible. On the other hand, magnitude bears strong connection to persistence as first demonstrated by Otter [Ott22]. Govc and Hepworth [GH21] further develop this concept by defining persistent magnitude for finite spaces, an invariant sharing many of the same properties held by magnitude, and suggesting a definition for compact spaces. We improve on this concept by introducing alpha magnitude, the persistent magnitude of an alpha complex of a metric space, and providing a definition for the alpha magnitude of compact metric spaces. Unlike magnitude, in cases like the one above in 2 dimensions, the computational complexity of alpha magnitude scales only linearly with the number of points.



Example 2: The alpha complex for the space from before, at different filtration levels. The barcode computed for the alpha complex is shown below.

We compute the alpha magnitude of the space X through the following equation over the barcode $\{[a_{k,i}, b_{k,i})\}_{i=1}^{m_k}$, where k is the homology degree and a and b are the endpoints of each bar. This equation is the general term for the persistent magnitude of a space X, an invariant first introduced by Govc and Hepworth [OKO22].

$$|X|_\alpha = \sum_{k=0}^{\infty} \sum_{i=1}^{m_k} (-1)^k (e^{-a_{k,i}} - e^{-b_{k,i}}) .$$

Alpha magnitude, like magnitude, is higher in scattered spaces and lower in concentrated ones. It approaches the cardinality of the space as the distances become very large, and approaches 1 when the distances become very small.

There are many definitions which exist to extend magnitude to compact spaces, but they are known to agree for positive definite compact metric spaces, by a result of Meckes [Mec13]. The most accessible is to take the magnitude of a positive definite compact metric space X to be

$$|X| = \sup_{\#(A)<\infty,\, A\subset X} |A|,$$

the supremal magnitude over all finite subsets of X. Similarly, for alpha magnitude, we take the alpha magnitude of a compact metric space to be

$$|X|_\alpha = \lim_{\#(A)<\infty,\, A\subset X} |A|_\alpha$$

when this limit exists over all finite sequences of subsets converging to X.

For positive definite compact metric spaces, the following expression is known to be equivalent to Minkowski dimension[Mec13]:

$$\dim_{Mink}(X) = \dim_{Mag}(X) = \lim_{t\to\infty} \frac{\log |tX|}{\log t}.$$

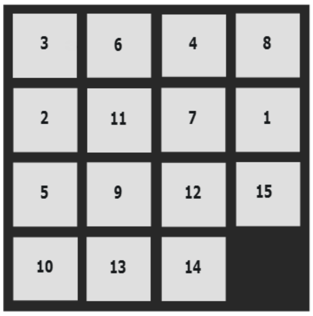
This limit is referred to as the magnitude dimension of the space X. While this result is initially encouraging for the purposes of dimension estimation via sampling, magnitude is much too expensive. Thus, we conjecture the following expression

$$\dim_\alpha(X) = \lim_{t\to\infty} \frac{\log |tX|_\alpha}{\log t}$$

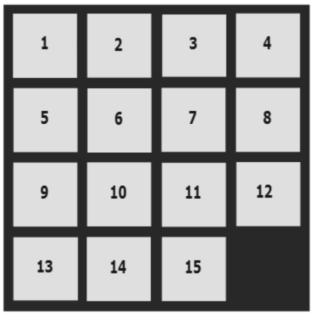
is equivalent to Minkowski dimension as well, where alpha magnitude exists. In the cases we examine, such as the Cantor set and the unit circle with the metric inherited from $\mathbb{R}^2$, we prove this to be true. For the Feigenbaum attractor, a set for which the Hausdorff dimension is only computationally approximated, our estimation falls quite close to existing methods. Thus, we posit that alpha magnitude dimension is a rich area to be examined in the interest of developing new means of estimating dimension.

Approximately optimal search

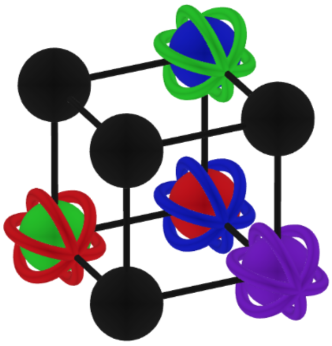
The higher-dimensional sliding puzzle we study is built on a d-dimensional cube (Each node of the cube is a binary string of length d, and two nodes are connected if they differ by a single bit) where $2^d - l$ randomly selected vertices are coloured (i.e., l vertices are uncoloured or have the same colour). Then, there are precisely $2^d - l$ rings with the same colours initially set randomly on vertices. The 15-puzzle can be seen as a version of this game played on a (4×4)-grid. In that setting, rings block each other's movement simply by being in the way. However, topologically, this can be considered to represent the rule that movement is blocked when the 1-simplex where the ring would move over is occupied by another ring. The higher-dimensional puzzle setting is a generalisation of this puzzle's setup as the (d, k, ℓ) scheme, where the puzzle is played on a d-dimensional cube with $2^d - \ell$ coloured vertices and rings, and where each move consists of moving one ring to a vertex which shared the same k-face so long as any other rings do not occupy that face [Wil74].



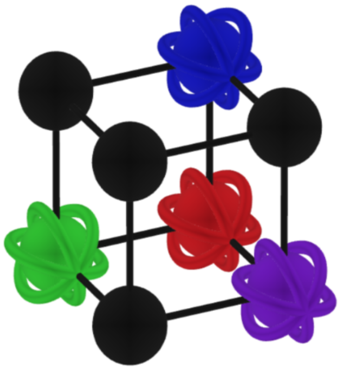
(a) 15-puzzle: Starting configuration.



(b) 15-puzzle: Target configuration.



(c) Cubical sliding puzzle: Starting configuration.



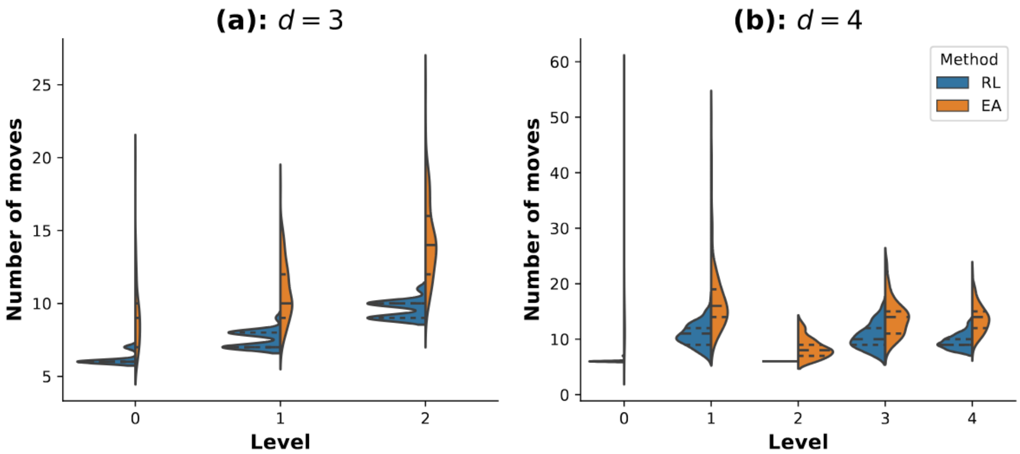
(d) Cubical sliding puzzle: Target configuration.

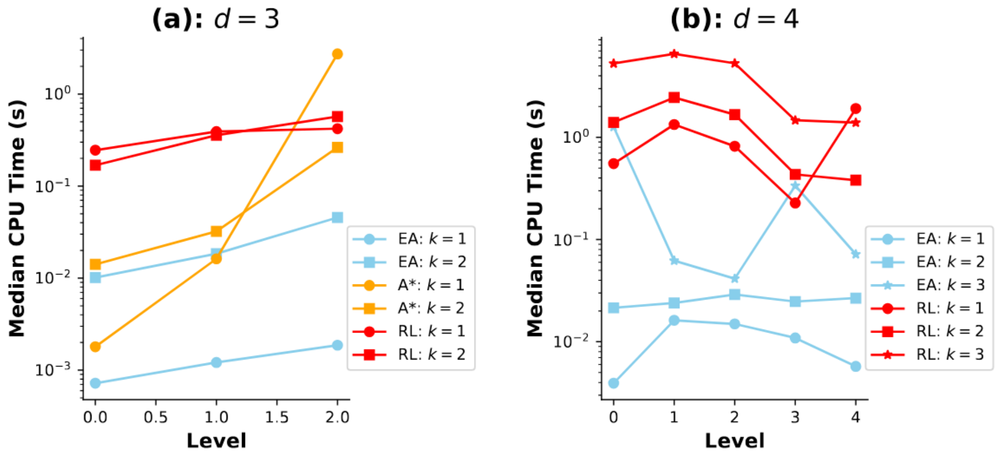
While in lower dimensions, optimal solutions of this sort of puzzle can be easily computed (e.g, 15 puzzle, lower dimensions in [BMRV23]), higher dimensional versions like the ones we examine in [MOMR24] are too expansive for this strategy to be viable. Further, existing methods of complete solutions making use of the typical blocking properties in multi-agent pathfinding problems only work for $k = 1$ in our above k-rule formulation, since rings can block movement across an entire face, rather than just their own vertex. As such, the methods we establish for solving the game are maximally general and can be applied to nearly any task requiring pathfinding on some form of connected configuration space.

First, we approach the problem with evolutionary algorithms. Fundamentally, an implementation of an EA involves the generation of multiple agents which independently work on the problem through random action. A selection force is then applied in a linear fashion to the population of agents, with agents performing better than others (determined by an objective function) more represented in the next generation of agents. A mutation operator is then applied to determine the nature of the next generation of agents, based on the previous one.

We also implement a reinforcement learning algorithm to solve the cubical sliding puzzle. For the purpose of this problem, we make two notable changes from the usual method of implementing RL in a pathfinding environment. First, we delay the assignment of rewards to the end of each search episode, since it is unclear whether particular moves are helpful or not until the game is over. Second, we implement an initial breadth-first search around the target configuration and assign initial weights to configurations around the target configuration based on how close they are to the target configuration. This process expedites the initial search the reinforcement learning algorithm performs, and provides optimal paths in proximity to the target configuration, improving the end result. While BFS is expensive, we perform this search for a very small number of configurations, so the CPU time expenditure is minimal.

Finally, we implement a version of A* search for the purpose of providing an optimal solution for this problem. A* search works by moving between adjacent nodes based on a combined cost function. The cost function for any given node x consists of a g-cost $g(C)$ representing the distance traveled to get to the node C from the starting configuration, and an h-cost $h(C)$ representing the estimated heuristic distance from node C to the target configuration. In the case of the sliding puzzle, our heuristic function simply counts the number of moves it would take to get to the target configuration from C if every node were completely unblocked.

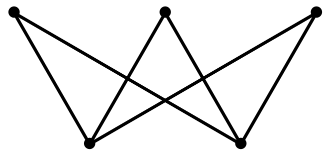




We observe that while the performance of our RL implementation is more consistent than the EA implementation, the RL algorithm is comparatively more expensive in terms of CPU time. In lower dimensions A* is efficient enough, however for puzzles such as the $d = 4$, $k = 3$ case, we note that A* fails to provide a solution in a reasonable timeframe. Both RL and EA are capable of providing solutions in a shorter period.

Stability of magnitude

In the case of magnitude, an important question remains the stability of magnitude for finite metric spaces. This is a critical result in the use of magnitude for data analysis. If small shifts in the underlying space can produce dramatically different magnitude, non-existent magnitude, or behavior of the magnitude which is in some sense degenerate, magnitude becomes less useful as a tool for the analysis of that dataset. Unfortunately, there are examples for which this is a very real possibility. Consider $K_{3,2}$, the bipartite complete graph on 3 and 2 points, endowed with the shortest path metric.



Example 3: The bipartite complete graph on 3 and 2 points, $K_{3,2}$

The magnitude function for this space is

$$|tK_{3,2}| = \frac{5 - 7\mathrm{e}^{-t}}{(1 + \mathrm{e}^{-t})(1 - 2\mathrm{e}^{-2t})}$$

and is thus undefined at $t = \log(\sqrt{2})$. We need to take care with cases like these. Leinster [Lei13] establishes that the magnitude function (a partially defined function obtained by introducing a scale factor t to a metric space (X, d)) is continuous at all but finitely many points in the space, and increasing for t sufficiently large. An existing result of Meckes[Mec13] establishes that magnitude is lower semicontinuous for positive definite finite metric spaces. In forthcoming work, we extend these results to a result for the continuity of magnitude for finite metric spaces of strictly negative type.

We say a metric space (X, d) is of strictly negative type if for any subset of $\{x_1, \ldots, x_n\} \subseteq X$, and all real numbers $\zeta_1, \ldots, \zeta_n \in \mathbb{R}$ with $\zeta_1 + \zeta_2 + \ldots + \zeta_n = 0$, we have

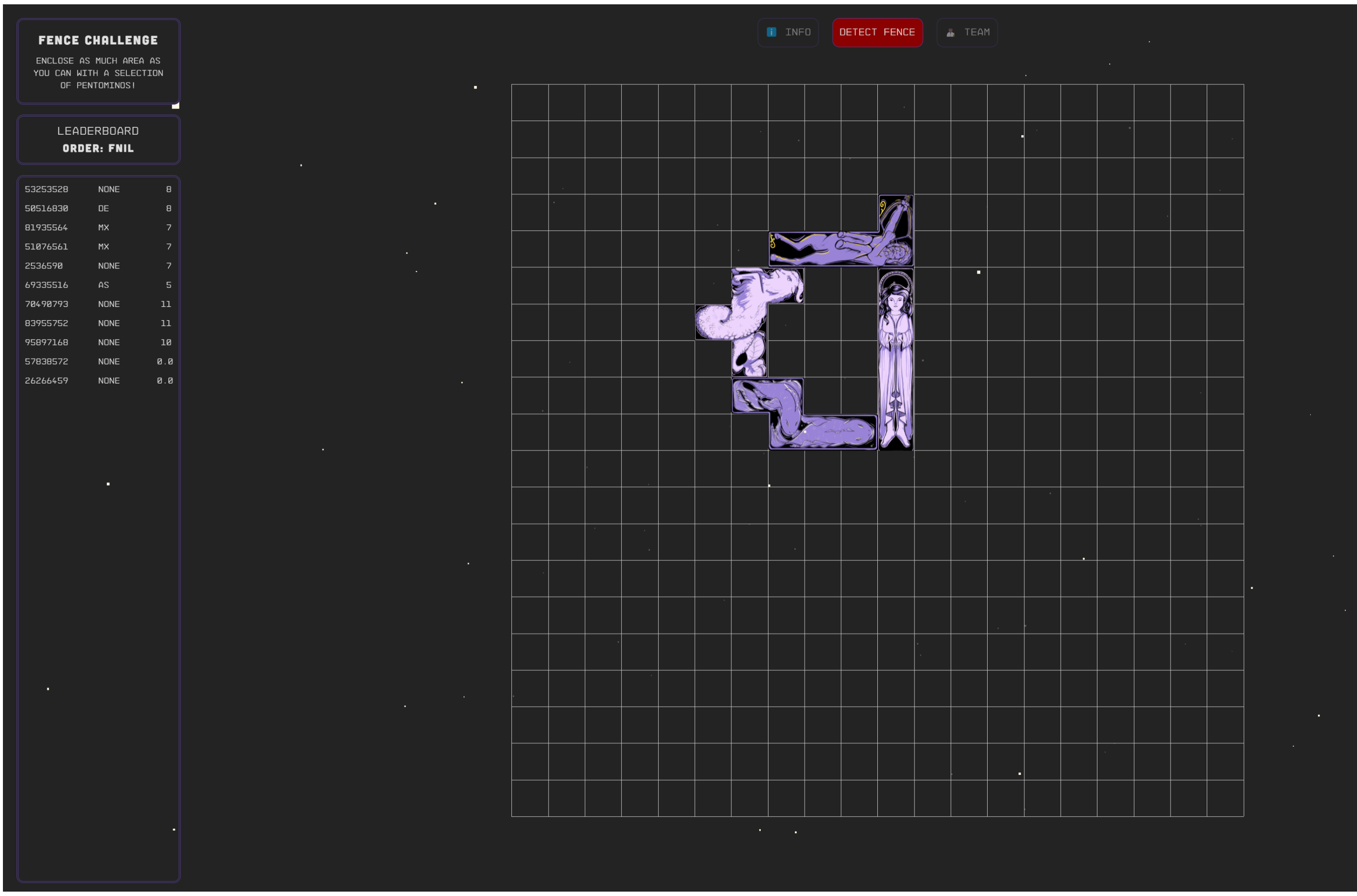
$$\sum_{1 \leq i \leq j \leq n} d(x_i, x_j)\zeta_i\zeta_j < 0.$$

For such spaces, (a class which includes all Euclidean spaces $\mathbb{R}^n$ with the usual metric) the magnitude function $|tX|$ is defined on $t \in (0, \infty)$ and can be extended to $t \in [0, \infty)$ in a continuous manner by defining $|tX| = 1$. This is desirable since magnitude is meant to provide an evaluation of the effective number of points in the space. However, if $\lim_{t \to 0} |tX| \neq 1$, we have a function which tells us a space with effectively one point has a different number.

The implications of a stability result of this nature for data science are substantial. So long as a point cloud is known to exist in a metric space of strictly negative type, the magnitude function will assuredly be continuous. Thus we achieve a wider class of datasets for which magnitude based clustering algorithms, dimension estimation, and other methods employing magnitude may be applied. In particular, finite data sets with irregular distance functions between observations become accessible so long as the point cloud is of strictly negative type.

Fence challenge and citizen science

Citizen science can be identified as the practice of involving non-researcher participants in the discovery of new and interesting results for various fields. In the case of mathematics, CS is dramatically underrepresented. We view this as unfortunate, since many areas (knots, combinatorial problems) have the potential to be worked on with minimal or no formal training. The fence challenge is a project we've undertaken to make a problem at the intersection of combinatorics and isoperimetric problems accessible to the broadest audience possible.



The goal of the game is to take the 12 pentominos, or a subcollection as shown above, and create as large of a fenced area as possible. This version of the isoperimetric problem allows players to develop their own strategies for optimizing the enclosed area, and helps in the development of spatial reasoning, as well as procedural problem solving. From the perspective of citizen science, we also argue this method is more fruitful than exhaustively searching all possible configurations, as there are simply too many possible orders of the 12 pentominos to search. Through a citizen science approach, we can learn about the distribution of possible fences, as well as human performance on similar problems involving optimization through spatial reasoning.[MO]

Future work

In future work, I would seek to further develop the theory of alpha magnitude and the means through which it may be employed in the service of data science. An open question is the stability of alpha magnitude in compact spaces, which would be a natural question. For connections to other invariants, magnitude is known to be related to curvature and volume, in addition to dimension. Persistent homology is also known to have strong connection to curvature, and so it would be reasonable to expect that alpha magnitude holds similar properties. Since alpha magnitude holds computational advantage over magnitude, results providing a connection to other invariants potentially provide a method to estimate these qualities with high fidelity.

Magnitude originally arose as a measure of biological diversity and thus bears strong correlation to clustering in metric spaces. Alpha magnitude has similar properties by construction, and thus warrants further study in this venue. I would develop algorithms which can be employed to use alpha magnitude to detect clusters. In datasets of low dimension, alpha magnitude can be computed in as little as linear time with respect to the cardinality of the dataset, again leveraging the computational advantage over magnitude. I would further seek to demonstrate the usefulness of such algorithms in comparison to existing methods of clustering. No clustering algorithm is perfect, but magnitude seems to be a particularly good one. It stands to reason that alpha magnitude can provide comparable results. Potential work could include identifying and classifying samples from natural fractal formations such as snowflakes or fungi, as well as other datasets which lend themselves well towards clustering analysis.

In addition, I would seek to employ magnitude in the estimation of dimension for datasets which alpha magnitude is ill-equipped. Since we have a stability result, we can approach this question with some confidence. In particular, magnitude is well suited towards computation for datasets of low cardinality but high dimension. This is precisely where alpha magnitude is weaker, since the computational speedup is only for datasets of lower dimension. Potential work could include medical datasets, where we have few individuals but numerous observations for different characteristics, survey data, and any other dataset where robust clustering analysis is desired and alpha magnitude is unsuitable.

The implementations of magnitude in clustering are manifold. A hierarchical clustering algorithm using magnitude is easily defined, simply by choosing clusters of lowest magnitude and working upwards. Leinster [Lei13] demonstrates that magnitude increases over expansions of Euclidean space, and our stability result assures that for spaces of negative type (such as Euclidean space) magnitude is continuous on $[0, \infty)$, so the use of magnitude in hierarchical clustering is appropriate. Another potential use of magnitude and alpha magnitude is as an easy check for other systems against existing clustering algorithms. A proposed cluster with particularly high magnitude is likely not very clustered at all, and so the use of magnitude provides a simple indicator as to whether a cluster ought to pass muster.

Another direction in which I'd take the development of alpha magnitude and magnitude is through implementation in machine learning environments. While magnitude is computationally inefficient in many cases, there are some (as described above, datasets of low cardinality but high dimension) where it makes sense. Others, such as Adamer et. al.[AODB$^+$21], have found success using the magnitude vector, a method which saves on computational expense by only considering local information. Such methods have potential to improve results for clustering through considering datasets in patches, thus allowing substantial computational speedup. I would further investigate the potential to employ such methods in the use of magnitude in data science, and seek to otherwise enhance the computational speed of magnitude.

On the other hand, since alpha magnitude is so easily computed, and yields such stark results in the case of the estimation of alpha magnitude dimension, it is reasonable to expect that ML applications of alpha magnitude could be achieved in time comparable to faster clustering algorithms, while preserving the richness of magnitude in the observations. The use of ML in clustering is well established, so the methods through which alpha magnitude may be implemented already exist and can be easily investigated to determine if there is improvement to be had. This direction of research has the potential to be exceedingly valuable, especially in the analysis of complicated data structures which defy traditional statistical methods of sorting. Topological data analysis has provided multiple such results in the past (cancer types [NLC11], diabetes [LCG+15], etc.) and so the use of alpha magnitude in similar settings is highly exciting.

Another subject I will pursue in future work is the extent to which ML algorithms, such as our RL and EA implementations for the higher-dimensional cubical sliding puzzle, can be applied to pathfinding settings in general. Any single-player game can be viewed as an instance of single or multi-agent pathfinding, since ultimately the goal of the player is to find the shortest route between their starting position and a "win" state, based on the moves they are allowed to make. I would further investigate optimal conditions for the initial breadth-first search implemented in our RL algorithm. Since the usage of some BFS outperforms both no BFS and entirely BFS, there is evidently an optimal amount of BFS for any particular problem where RL can be implemented with a known target configuration. Determining where this point of optimality lies, even roughly, would prove extremely promising for any algorithm implementing ML techniques in the service of search.

Publications

[AODB+21] Michael Adamer, Leslie O'Bray, Edward De Brouwer, Bastian Rieck, and Karsten Borgwardt. The magnitude vector of images, 10 2021.

[BMRV23] Moritz Beyer, Stefano Mereta, Érika Roldán, and Peter Voran. Higher-dimensional cubical sliding puzzles. *arXiv Preprint arXiv:2307.14143*, 2023.

[GH21] D. Govc and R. Hepworth. Persistent magnitude. *Journal of Pure and Applied Algebra*, 225, 2021.

[LCG+15] Li Li, Wei-Yi Cheng, Benjamin S. Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P. Bottinger, and Joel T. Dudley. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine*, 7(311):311ra174–311ra174, 2015.

[Lei13] Tom Leinster. The magnitude of metric spaces. *Documenta Mathematica*, 18:857–905, 2013.

[Mec13] Mark W. Meckes. Positive definite metric spaces. *Positivity*, 17(3):733–757, 2013.

[MO] Erika Roldan Miguel O'Malley. On the fence challenge and citizen science.

[MOMR24] Nono CM Merleau, Miguel O'Malley, Sayan Mukherjee, and Erika Roldan. Approximately optimal search on a higher dimensional sliding puzzle., 12 2024.

[NLC11] Monica Nicolau, Arnold Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences of the United States of America*, 108:7265–70, 04 2011.

[OKO22] Miguel O'Malley, Sara Kalisnik, and Nina Otter. Alpha magnitude, 05 2022.

[Ott22] Nina Otter. Magnitude meets persistence. Homology theories for filtered simplicial sets. *Homology, Homotopy and Applications*, 24:401–423, 2022.

[Wil74] Richard M Wilson. Graph puzzles, homotopy, and the alternating group. *Journal of Combinatorial Theory, Series B*, 16:88–96, 1974.