Persistent Homology as a Tool to Improve Jet Clustering and Analysis

Noah Green
PHY 493
Michigan State University

April 27, 2016

Abstract

As particle accelerators that operate at ever higher energies are built, it will be necessary to develop techniques that can get meaningful data out of highly boosted jets. Persistent homology is a relatively new technique that can be used to determine the structure of point clouds, and has the potential to improve current jet analysis techniques. It has the ability to distinguish the distance scales on which different structures can be resolved. This could potentially be used to define a variable jet clustering radius that could provide an alternative to the current large-R jets + substructure variables method of analyzing highly boosted jets. However, methods to analyze large numbers of persistence diagrams need to be developed prior to making any conclusions.

1 Introduction

Persistent homology is a powerful mathematical tool that is still maturing in the world of topological data analysis. It can be used on any data that can be made into a point cloud. The mathematics behind persistent homology will be covered in section 2, but it is essentially a technique to find the number of n-dimensional holes that stick around the longest as the points are systematically connected to each other. One area in physics where this could possibly be useful is in the analysis of particle jets.

Jets are the signature of interactions involving gluons and quarks (a.k.a. QCD interactions) in today's particle accelerators. Say that a quark and antiquark are formed in the center-of-momentum frame. Initially, they move apart from each other as free particles with equal and opposite momentum. However, due to asymptotic freedom, the strong force between them becomes incredibly strong as the quarks get about a femtometer from each other. At this point, if the quarks have enough energy, it becomes energetically favorable - as opposed to the quarks to continue pouring more energy into the strong force bond between them as they move apart - for new quark-antiquark pairs to be produced. This can happen many times over, with the new quark-antiquark pairs combining to form a shower of mesons and baryons. This process is called *hadronization*, and the resulting shower of particles is called a *jet*.

Working backwards and finding out what formed a jet not only allows for the identification of the original quark/gluon that caused it, but also works as a test of our knowledge of how the strong force works. Calculating the persistent homology on a collision that produces jets could help develop better algorithms to cluster particles into jets. Some current jet clustering and analysis techniques are discussed in section 3. A proof-of-concept demonstration and suggestions to improve current jet clustering techniques with persistent homology are proposed in section 4.

2 Persistent Homology

Before we dive into persistent homology, a few mathematical concepts are needed.

- A **simplex** is a generalization of a triangle. Hence, a 0-simplex is a point, a 1-simplex is an edge, a 2-simplex is a triangle (including its face), a 3-simplex is a tetrahedron (including the points inside). The analogy continues for higher dimensions, but the precise

mathematical definition is not required for understanding. Note that the faces of a simplex are themselves simplices of a lower dimension.

- A **simplicial complex** is a collection of simplices, K, so that

  1. If $\sigma \in K$ and τ is a face of σ, then $\tau \in K$.
  2. If σ_1,$\sigma_2 \in K$, then $\sigma_1 \cap \sigma_2$ is either empty or a face of both.

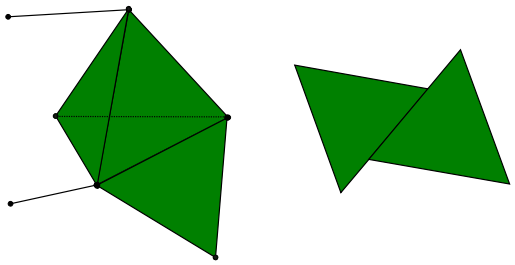

Figure 1: The shape on the left is a simplicial complex; the shape on the right is not since the triangles do not entirely share an edge.

  A simplicial complex can be expressed abstractly by numbering its vertices, and listing the vertex labels for each simplex in the complex. For example:

  - A point: $\{1\}$
  - An edge: $\{\{1,2\},\{1\},\{2\}\}$
  - A triangle: $\{\{1,2,3\},\{\{1,2\},\{2,3\},\{1,3\},\{1\},\{2\},\{3\}\}$

- Let K be an abstract simplicial complex, and $K^{(0)}$ be its vertex set. An **orientation** on the simplices of K is a partial order "$<$" on on $K^{(0)}$ so that "$<$" restricted to each simplex $\sigma \in K$ is a total order on σ. This means that a simplicial complex is oriented if all of the vertices in each simplex can be ordered from least to greatest (the ordering can be chosen arbitrarily), but simplices that do not share a face have their own unique ordering.

- Let K be a finite oriented abstract simplicial complex. The **group of n-chains on K**, $C_n(K;\mathbb{Z})$, is the set of functions $\varphi : K^{(n)} \setminus K^{(n-1)} \to \mathbb{Z}$. Note that $K^{(n)} \setminus K^{(n-1)}$ is just the set of n-simplices inside a simplicial complex. The purpose of this construction is to let us do algebra with the simplices of a simplicial complex by associating an integer with each of them.

- Let $\sigma \in K^{(n)} \backslash K^{(n-1)}$ where $\sigma = u_0, u_1, \ldots, u_n$ is a simplex with vertices u_i. The **boundary operator** is a function, $\partial_n(\sigma) : C_n(K;\mathbb{Z}) \to C_{n-1}(K;\mathbb{Z})$, defined as

$$\partial_n(\sigma) = \sum_{j=0}^{n} (-1)^j \{u_0, \ldots, \hat{u_j}, \ldots, u_n\}$$

  where $\hat{u_j}$ denotes that the j^{th} vertex has been removed. The boundary operator, ∂_n is important because it ends up being equal to zero when applied to chains that do not describe the boundary of an n^{th} order simplex in $K^{(n)} \setminus K^{(n-1)}$. This leads us to our next definitions.

- A chain $\varphi \in C_n(K;\mathbb{Z})$ is

  1. A **cycle** if $\partial_n(\varphi) = 0$. That is $\varphi \in ker(\partial_n) := Z_n(K;\mathbb{Z})$.
  2. A **boundary** if $\varphi \in Im(\partial_{n+1}) := B_n(K;\mathbb{Z})$.

With the proper language in place, we can now get to homology. The n^{th} homology group of a simplicial complex K with coefficients in $\mathbb{Z}$ is given by

$$H_n(K;\mathbb{Z}) = Z_n(K;\mathbb{Z})/B_n(K;\mathbb{Z}),$$

or in other words, it is the group of cycles modulo the group of boundaries. The dimension of the homology group is a direct reflection of the number of holes that a simplicial complex has. For example, if we stretched a circle, S^1, into a triangle then calculating the homology would pick up the 1-cycle going around the triangle to give $H_1(S^1;\mathbb{Z}) = \mathbb{Z}$. If we did the same process to a figure-eight, then it would pick up two 1-cycles to give $H_1(\text{figure-eight};\mathbb{Z}) = \mathbb{Z} \oplus \mathbb{Z}$.

The full algorithm to find the homology of a simplicial complex is beyond the scope of this paper, but it can be summarized.

1. Calculate the $(n+1)^{th}$ boundary operator as a matrix. This can be done by having the rows represent the n-simplices in the simplicial complex, and columns that have the coefficients of the $(n-1)$-simplices that come from applying the operator.

2. Reduce the matrices to Smith-normal form using row and column operations. Smith-normal form is achieved when the only non-zero terms are on the diagonal, and the $M_{i,i}^{th}$

diagonal divides the $M^{th}_{i+1,i+1}$ term on the diagonal. Keep track of the linear combinations of simplices that are formed by each row/column operation during this process.

3. There is a useful theorem for $\varphi \in C_{n+1}(K;\mathbb{Z})$ which says that $\partial_n(\partial_{n+1}(\varphi)) = 0$. Hence, the generators of the kernel of ∂_n and the image of ∂_{n+1} can just be read off from the Smith-normal form matrix. By taking the generators of the kernel of ∂_n modulo the generators of the image of ∂_{n+1}, the n^{th} homology group of the simplicial complex is recovered.

The persistence part of persistent homology requires one additional concept. A **filtration** of a simplicial complex K is a sequence $\emptyset = K_0 \subset K_1 \subset K_2 \subset \ldots \subset K_{N-1} \subset K_N = K$ such that each K_i is a simplicial complex. An example is the Rips filtration, $R_\varepsilon(x)$, where given a set X and a distance metric $\rho : X \times X \to [0, \text{inf})$ and $\varepsilon > 0$

$$R_\varepsilon(x) = \{\{x_0, \ldots, x_n\} | \rho(x_i, x_j) < \varepsilon, 0 \leq i, j \leq n\}.$$

In other words, the Rips filtration is given by the simplicial complexes that are generated when all points inside spheres of radius ε at every point are connected. Naturally, a **filtered** simplicial complex is a simplicial complex partnered with a filtration. The persistence of a certain homology on K is given by the number of sub-complexes that display that homology.

The persistence of homologies in a filtered simplicial complex can be represented visually by a "barcode" that keeps track of the homologies of each sub-complex. When a new homology appears, it is said to be *born*, and when it disappears, it *dies*. As can be seen in figure 2, a

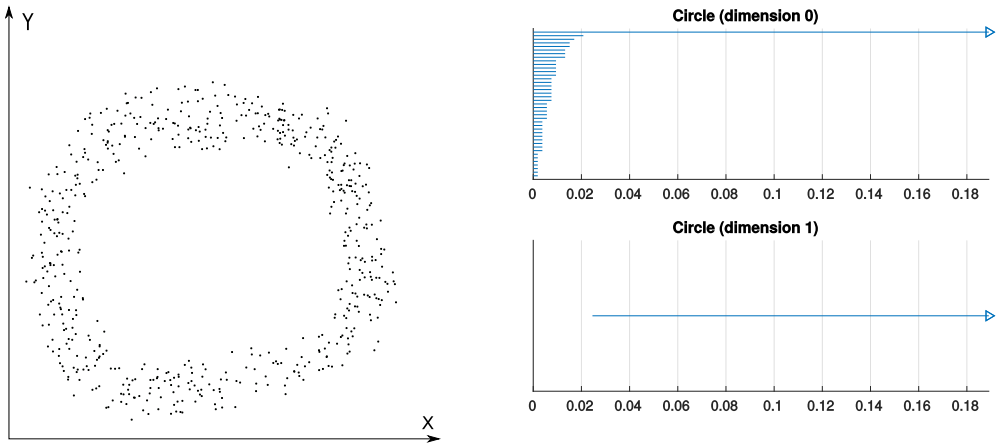

Figure 2: A circular point cloud, like that on the left, generates a barcode with only one significant bar in H_0 and H_1. Barcode was generated using *javaplex*[9].

point cloud with a circular distribution will start out with nothing in H_1, and many bars in H_0.

This is because H_0 is picking up each small group of points as they are clustered together, but there are not any one-dimensional holes formed during this early clustering for H_1 to pick up. However, say that the Rips filtration is being used. Once $\varepsilon \approx 0.025$, all of the points have been connected to enclose the hole in the middle of the circle, so a bar for H_1 is born, and all of the bars for H_0 die except one since there is now only one grouping of points.

3 Jet Analysis

Jets are seen in particle accelerators as a roughly grouped set of particles with momenta aligned in a similar direction. Currently, researchers have a few different ways to determine how to cluster these particles so that they correspond with the original quark or gluon that spawned them. The most successful of these techniques has been the sequential jet clustering algorithms. These algorithms all work very similarly, with the main difference being their treatment of the weighting of the distance measure by momentum. The general algorithm goes as follows[4]:

1. Take a pair of particles i, j and calculate the distance measure between them

$$d_{ij} = \min(p_{ti}^{2l}, p_{tj}^{2l})\Delta R_{ij}^2/R^2$$

and the distance measure between particle i and the beam

$$d_{iB} = p_{ti}^{2l},$$

where p_t is the component of the momentum of a particle that is perpendicular to the beam (a.k.a. the transverse momentum), and R is the jet-radius parameter, usually taken to be about 1. $\Delta R_{ij}^2 = (\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2$ is the distance between the particles in azimuth-pseudorapidity space. Note that pseudorapidity is defined as $\eta = -\ln\left[\tan(\frac{\theta}{2})\right]$.

2. Find the minimum distance measure out of all d_{ij} and d_{iB}. If the minimum is a d_{ij}, merge the particles by summing their four-momenta. If the minimum is a d_{iB}, then particle i is declared a jet and removed from further clustering.

3. Repeat these steps until no particles are left.

The type of clustering algorithm depends of the value of l:

- Kt algorithm: $l = 1$
- Cambridge-Aachen(CA) algorithm: $l = 0$
- Anti-kt algorithm: $l = -1$

Each algorithm has its own advantages and disadvantages, and their use depends on how momentum should be treated for a particular analysis. In general, the anti-kt algorithm is used if circular jets are desired, while the CA algorithm is good for looking at jet substructure.

One of the biggest problems in jet clustering arises when a massive relativistic particle is produced. In their rest frame, if these particles decay into two jets, the jets will emerge back-to-back. However, since they are highly relativistic, or boosted, the jets can emerge from the decay nearly on top of each other. We cannot throw out events with such particles because in many cases, they are exactly what we are looking for. As we look for new physics, we will be interested in the properties of heavy particles such as top quarks, and heavy particles that are theorized only to exist at high energies.

A common technique to deal with this problem is to perform an analysis on how the jets were clustered. Typically, the CA clustering algorithm is used to construct jets with a large radius. The clustering on these large-R jets is then run backwards to pick out the major subjets – the four-momentum sum of particles grouped together by the clustering algorithm – that were clustered last. The positions and momenta of these subjets can then be run through a substructure variable that is sensitive as to whether or not the subjets are spread out like the top quark event in figure 4. See reference [10] for an example.

4 Persistent Homology for Jets

Persistent homology probably won't work to replace clustering algorithms. The sequential clustering algorithm works very quickly, with computation time increasing quadratically with the number of particles in an event. However, computing the persistent homology involves reducing a boundary matrix to Smith-normal form, where computation time increases quartically with the number of simplices. Since high energy physics analyses often deal with millions of events at once, calculating the persistent homology of all of them would be impractical, unless a more efficient algorithm is found.

However, persistent homology may be useful in improving jet clustering algorithms. Computing it for an event reveals the distance scale at which structures and substructures of the event live. Hence, the persistent homology of events may be able to be used to define a variable jet radius for clustering algorithms based on some other quantity, such as their energy or transverse momentum. Doing so could provide a viable alternative to the current jet substructure methods. This bring up the question: How well does persistent homology interpret jet structures?

One situation where this can be tested is in distinguishing top jets from regular QCD jets. Top quarks have an extremely short lifetime, so when they decay into jets, the daughters are still relatively close to the original collision vertex.

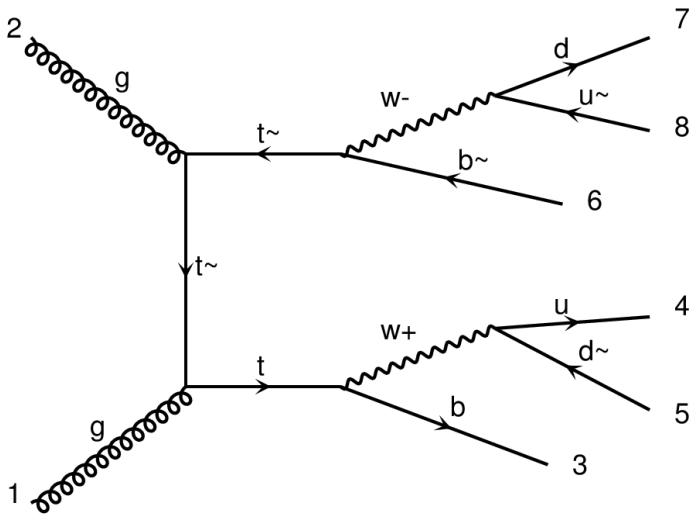

Figure 3: Top quarks can decay quickly into three lighter quarks, each of which form a jet. Diagram made by Madgraph5_aMC@NLO.

This means that each top quark is distinguished by having three separate jets that exit the event next to each other. Other QCD particles (quarks and gluons) will not decay so quickly, so their jets will not have these three distinct prongs.

To look at such events, the following procedure was used to generate their point clouds and calculate the persistent homology:

1. The Madgraph Monte-Carlo event generator was used to generate $t\bar{t} \to jets$ events and QCD jet events separately [3]. Pythia [8] was used to calculate the hadronization of these particles into jets.

2. Events were "cleaned" so that the major jet features were more prominent. (1) Only particles with no daughters were kept, since these are the particles that would be seen in

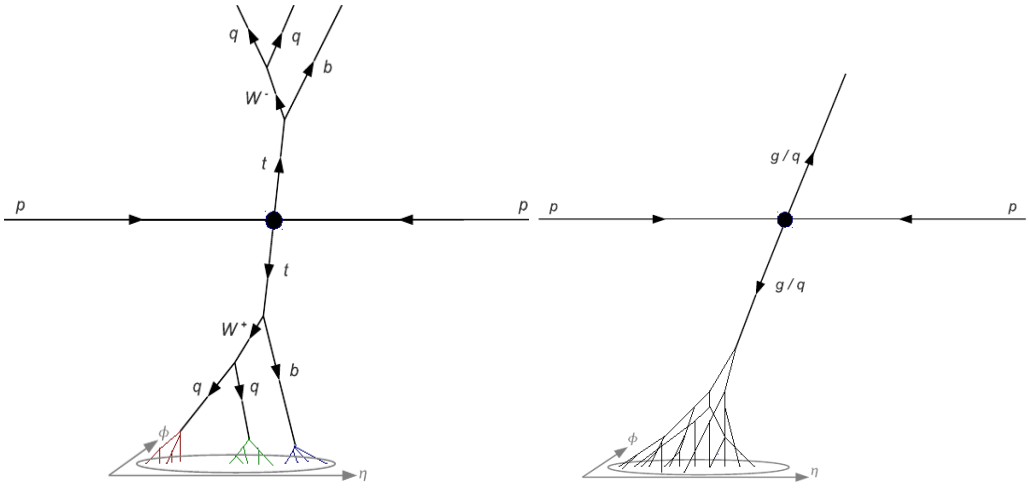

Figure 4: Top quarks decaying to jets have three prongs (left), while QCD jets with invariant mass near that of the top quark can have fewer prongs (right). Figure from reference [10]

the detector. (2) The $\eta - \phi$ distance between each pair of particles was arranged into a distance matrix. (3) Particles with $\eta > 4.0$ were thrown out since they would be nearly parallel with the beam. (4) Isolated particles where $\Delta\eta^2 + \Delta\phi^2 > 4.0$ with their nearest neighbor were thrown out since they would not be clearly in any jet.

3. The persistent homology was calculated on the remaining distance matrix.

The Rips filtration was used for calculating the persistent homology because it is sensitive to the structure of the denser areas of the point cloud. Initially, *javaplex* [9] was used for calculating the homology, but it was too computationally expensive to use for iteration (For the events that were calculable, it made simplicial complexes with $> 10^6$ simplices.). Instead, *TDATools* [11] was used. This software uses a Rips-collapse algorithm that generates the same diagrams as the Rips filtration, but is considerably faster. It has the disadvantage of only being able to calculate up to H_1, so only the two-dimensional $\eta - \phi$ space was used in building the event point clouds.

Typical results for events are shown in figures 5 and 6. They are formatted as *persistence diagrams*. Note that these carry the same information as the barcode example of section 2, only the birth of a feature is represented on the horizontal axis, while the death of a feature is represented on the vertical axis. Hence, the most significant features of the point cloud will be those farthest from the diagonal.

H_0 shows much more structure for the $t\bar{t} \rightarrow jets$ event, with ≈ 6 subjets showing up as expected. The QCD jet event has two main features for H_0, as expected for two jets leaving the collision back-to-back. Note that the point at -1 represents the point at infinity; this is due to

H_0 picking up the entirety of each point cloud. H_0 also appears to be useful in finding the jet radius, as desired. There is a clear gap between the deaths of the initial clustering of particles, and the deaths of the jets/subjets.

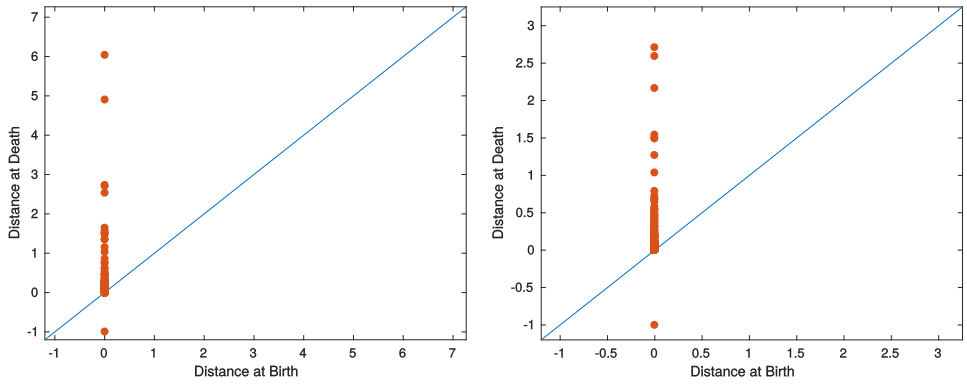

Figure 5: Persistence diagram for H_0 for a QCD jet event (left) and a $t\bar{t} \to jets$ event (right).

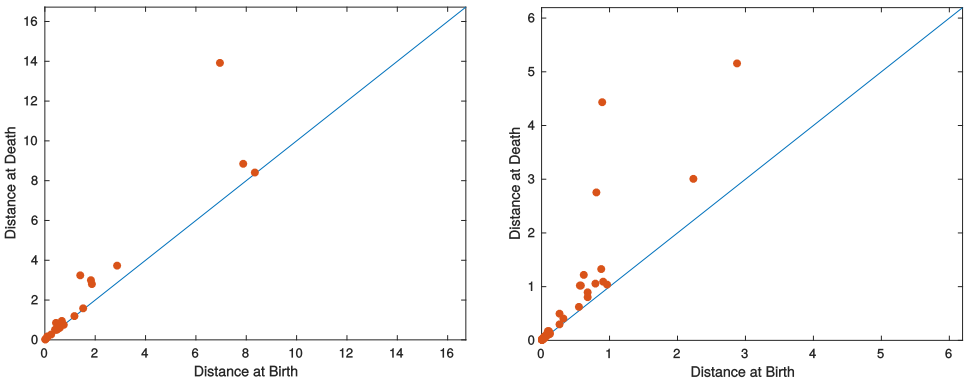

Figure 6: Persistence diagram for H_1 for a QCD jet event (left) and a $t\bar{t} \to jets$ event (right).

H_1 ended up having more features than expected. It is possible that it is picking up the "hole" that is formed between the prongs of each jet as the Rips distance is increased. This would be a good way to distinguish jets with ≥ 3 prongs if that is the case.

The next step would be to develop a metric to measure the distance between persistence diagrams for many events. This would allow us to verify that QCD jet and top jet events do indeed segregate into unique types of diagrams, and allow us to see if there is a pattern in the jet radius as a function of some other variable.

5 Conclusion

It is important to understand jets because they are the dominant feature that is detected in the high-energy particle accelerators of today and tomorrow. Their simulation and analysis not only helps us understand what particles produced them, but also serve as a test of our knowledge of how QCD works. A jet is seen in a detector as a group of collimated particles. In order to recover the properties of the original quark or gluon that produced it, these particles must be clustered together as seen in section 3.

As accelerators collide particles at ever higher energies, they will produce highly boosted standard model (SM) particles and perhaps more massive particles the will decay into highly boosted SM particles. If these particles decay into jets, then instead of being back-to-back, as in the rest frame, the highly boosted frame will cause them to enter the detector nearly on top of each other. This problem is dealt with today by clustering particles within a larger radius for highly boosted jets, and using the resolution of the detector to probe the substructure of the jet as seen in section 3.

These considerations show that accelerator technology, not considering cost, will continue to be limited by our ability to analyze and detect jets. Persistent homology may be a useful tool in finding ways to improve current clustering algorithms and jet substructure tools by allowing a better understanding of jet size and structure as a function of other variables, such as energy or momentum. It was seen in section 4 that it can be used to interpret jet structure, but methods to analyze a larger sample size need to be developed before making any conclusions.

Bibliography

[1] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.

[2] Georges Aad et al. Search for the $b\bar{b}$ decay of the Standard Model Higgs boson in associated $(W/Z)H$ production with the ATLAS detector. *JHEP*, 01:069, 2015.

[3] Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. MadGraph 5 : Going Beyond. *JHEP*, 06:128, 2011.

[4] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J.*, C72:1896, 2012.

[5] Robert Ghrist. Barcodes: the persistent topology of data. *Bull. Amer. Math. Soc.*, 45:61–75, 2008.

[6] David Griffiths. *Introduction to Elementary Particles*. WILEY-VCH Verlag GmbH and Co., 2010.

[7] John L. Harer and Herbert Edelsbrunner. *Computational Topology: An Introduction*. The American Mathematical Society, 2010.

[8] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. PYTHIA 6.4 Physics and Manual. *JHEP*, 05:026, 2006.

[9] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. JavaPlex: A research software package for persistent (co)homology. In Han Hong and Chee Yap, editors, *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pages 129–136, 2014.

[10] Jesse Thaler and Ken Van Tilburg. Identifying Boosted Objects with N-subjettiness. *JHEP*, 03:015, 2011.

[11] Chris Tralie. User's guide for tdatools package. *http://www.ctralie.com/Teaching/Math412_F2014_MusicAssignment/TDATools_UsersGuide.pdf*, 2014.