Hyperparameter Optimization of Topological Features for Machine Learning Applications

Francis Motta
Dept. of Mathematical Sciences
Florida Atlantic University
Boca Raton, FL, USA
fmotta@fau.edu

Christopher Tralie
Dept. of Mathematics and Computer Science
Ursinus College
Collegeville, PA, USA
ctralie@ursinus.edu

***Abstract*—This paper describes a general pipeline for generating optimal vector representations of topological features of data for use with machine learning algorithms. This pipeline can be viewed as a costly black-box function defined over a complex configuration space, each point of which specifies both how features are generated and how predictive models are trained on those features. We propose using state-of-the-art Bayesian optimization algorithms to inform the choice of topological vectorization hyperparameters while simultaneously choosing learning model parameters. We demonstrate the need for and effectiveness of this pipeline using two difficult biological learning problems, and illustrate the nontrivial interactions between topological feature generation and learning model hyperparameters.**

***Index Terms*—hyperparameter optimization, topological data analysis, persistence diagrams, machine learning**

I. Introduction

Topological data analysis (TDA) tools are increasingly being combined with machine learning (ML) methods to both improve the accuracies of predictive models and gain scientific insights. Broadly speaking, TDA tools transform data into quantified, interpretable characterizations of the data's latent geometric structure. In particular, *persistent homology* (PH) transforms data into *persistence diagrams* (PDs), which are quantified representations of topological structures, encoded as collections of *persistence pairs* (PPs) in the plane.

To take advantage of the host of well-established learning models for classification and regression problems, researchers have proposed numerous methods to further transform PDs into vector representations that are amenable to ML. For example, one popular algorithm transforms a PD into a so-called *persistence image* (PI) by first generating a surface over the plane by convolving the PPs with a chosen kernel, and then integrating the surface over each "pixel" defined by a chosen gridding of the plane [1]. This particular embedding of the diagram into a finite-dimensional vector space involves several choices including the resolution of the image and the bandwidth and scaling of the kernels. The authors of [1] demonstrate empirically that eventual model performance can be insensitive to the choice of these "vectorization" parameters. However, the degree to which this is true is problem-specific and may also depend on the choice of model assessment.

DARPA Synergistic Discovery and Design

Even simple PD vectorization approaches involve choices. For example, in [2], the authors studied the linear relationships between brain arterial age and topological feature vectors derived by first sorting PPs according to their persistence and then encoding the PP coordinates between the kth to the jth most persistent pairs into a length-$2(j-k+1)$ vector, for many choices of j and k. The authors observe that the accuracy of the models depend in a surprising way on the choice of j and k, which illustrates that medium-persistence topological features are correlated with age, and thus provides insight into the way arterial structures change over time.

It is well known, and well accepted in the machine-learning community that the accuracy of a learning model's predictions may depend on choices of tunable hyperparameters (HPs). For example, the hyperparameter *minimum samples per leaf* (MSL) sets a lower bound on the number of samples required to be in a leaf node of a decision tree and thereby ensures that a the leaves do not contain too many points to be discriminating nor too few to be generalizable.

The problem of finding such an optimum is made especially difficult by the large number of tunable HPs in most modern machine-learning methods, and the high computational cost often required to train these learners. Chief among the proposed solutions to the problem of learning-model HP optimization is Bayesian optimization—a framework developed to efficiently optimize computationally expensive black-box functions defined over complex parameter spaces.

II. Hyperparameter Optimization

A supervised learning algorithm endeavors to learn a representation of a functional relationship between a set of input variables (predictors/features) and output variables (targets) from a finite collection of samples of the underlying input and target spaces. Numerous methods exist for modelling such an unknown relationship [3] and most of these models are highly flexible, as they contain many parameters which are fit to available data through some prescribed training procedure. Indeed, it is often possible with such a model to construct a perfect representation of the unknown function that has little utility in practice due to overfitting. To avoid the problem of overfitting, most supervised learning methods are equipped with HPs controlling the maximum complexity of the model, such as additional regularization parameters [4]. Moreover, most ML methods require choices of the particular functional forms that serve as the building blocks of the model's representation of the unknown function, e.g. the choice of kernel in a nonlinear support vector machine [5]. Notably, in general, the collection of model HPs defining the *configuration space* (CS) of a ML algorithm may include combinations of discrete and continuous variables as well as conditional dimensions that depend on the choice of other HPs [6].

To each configuration of a learning model one can associate a model performance metric—usually a measure of the accuracy of a trained model's predictions on data not used to fit the model. In this way, one defines a black-box function over the configuration space, whose evaluation involves training and testing the supervised learner. The practical problem is to locate the best choice of HPs which minimize the error in model predictions.

A. Bayesian Optimization

The most promising methods for optimizing a costly black box function, f, are Bayesian optimization approaches [6]–[8]. The advantages of these approaches in the context of HP optimization is that they limit the number of evaluations of f by introducing a prior *surrogate function* that approximates f and that is designed to be much faster to evaluate than f. In this way, an informed recommendation regarding which configuration should be next evaluated using the costly function f can be made (via an *acquisition function*). This iterative process then updates the prior surrogate function with this new information to yield an improved (posterior) approximation f. Furthermore, these methods are well suited to optimize unknown functions over complex, even conditional CSs [9].

In this work, we apply and make extensive use of the Python module Hyperoprt and take advantage of its implementation of so-called *choice nodes* that encode a list as the values of a discrete random variable [10]. These categorical spaces allow us to express conditional parameters that depend on other choices of parameters [11] which enables simultaneous exploration of the performance of multiple PD vectorization methods.

III. Topological Data Analysis

A. Persistent Homology

Homology is a computational device that can be used to count the numbers of components, holes, voids, and higher-dimensional analogues of voids of a space (See [12] for a full treatment). PH extends homology by capturing the homological structure of a nested family of topological spaces, parameterized by a real interval. Although homology and PH may be defined more generally, we restrict the construction to simplicial homology in service of the applications discussed in Sections IV-B and IV-A.

1) Simplicial Complexes: A simplicial complex is a combinatorial representation of a topological space that may be regarded as a generalization of a graph that may contain higher-dimensional analogues of vertices and edges, called *simplices*. In particular, given a finite set of points $\{v_1, \ldots, v_N\}$, a k-simplex, $\sigma = \{v_{i_0}, \ldots, v_{i_k}\}$, is subset of $k+1$ vertices. In the same way that vertices of a graph (now 0-simplices) are singleton subsets, and edges (now 1-simplices) are subsets of size two, higher dimensional simplices are defined as larger subsets, e.g. 2-simplices (faces) are subsets of size three, 3-simplices (tetrahedra) are subsets of size four, etc. A collection of simplices $\mathcal{S}$ that satisfies

- if σ is a simplex in $\mathcal{S}$, then $\mathcal{S}$ also contains all subsets of σ, and
- the intersection of any two simplices in $\mathcal{S}$ is either empty or is a simplex in $\mathcal{S}$

is called a *simplicial complex*.

2) Simplicial Homology: The presence of three boundary edges $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$ in $\mathcal{S}$ that are associated with a 2-simplex, $\sigma = \{a, b, c\}$ do not guarantee the existence of σ in $\mathcal{S}$. Simplicial homology aims to reveal such holes using a formal algebraic construction. Say that within a simplicial complex $\mathcal{S}$, the number of k-simplices is n_k. Then an n_k-dimensional vector space, $C_k(\mathcal{S})$ over the field, $\mathbb{F} = \mathbb{Z}/2\mathbb{Z}$, may be defined to be the collection of all formal sums of the k-simplices. For each k, there is a natural linear map $\partial_k : C_k \to C_{k-1}$ which maps a k-simplex to its boundary, represented as a linear combination of the boundary $(k-1)$-simplices. A linear combination of k-simplices which is in the image of ∂_{k+1} is the *k-boundary* of the collection of $(k+1)$-simplices, while a *k-cycle* is a linear combination of k-simplices which map to 0 under ∂_k. Because $\partial_k \circ \partial_{k+1} = 0$, boundaries are always cycles although there may be cycles which are not boundaries. To identify all the cycles which are not boundaries, one computes the *k-th order homology group* as the quotient of vector spaces, $H_k(\mathcal{S}) := Z_k(\mathcal{S})/B_k(\mathcal{S})$, which consists of equivalence classes of k-cycles in which two k-cycles are equivalent if they differ by a k-boundary.

3) Nested Complexes, Persistent Homology, & Diagrams: A filtered family of simplicial complexes parameterized by an interval is a collection of simplicial complexes $\mathcal{S}_r$ for $r \in [s, t]$ such that if $x \leq y$ then $\mathcal{S}_x \subseteq \mathcal{S}_y$. One imagines starting with the complex, $\mathcal{S}_s$, and adding to it simplices as r increases from s to t. A simplex σ will be said to appear at $r = y$ if $\sigma \in \mathcal{S}_y$ and $\sigma \notin \mathcal{S}_x$ for any $x < y$.

Let $\{\mathcal{S}_r\}_{r\in[s,t]}$ be a filtered family of simplicial complexes. The *kth order PH groups* of $\{\mathcal{S}_r\}_{r\in[s,t]}$ are the vector space quotients

$$H_k^{x,y} := Z_k(\mathcal{S}_x)/(B_k(\mathcal{S}_y) \cap Z_k(\mathcal{S}_x)),$$

for $x \leq y$. For each $s \leq x \leq y \leq t$, the dimension of $H_k^{x,y}$ counts the number of k-holes which first appeared in some complex $\mathcal{S}_r \subseteq \mathcal{S}_x$ and which are still present in $\mathcal{S}_y$ (See [13], Chapter VII.1).

For each homological dimension k, the kth order PH groups are faithfully represented by the kth order PD, consisting of finitely many pairs (b, d) with $b \leq d$ that specify the filtration parameters at which each k-hole appears/is born (b) and disappears/dies (d).

4) Filtrations on Point Clouds: To capture the intrinsic homological structure of a point cloud, it is common to construct a filterered family of simplicial complexes whose vertices are the points in the cloud. Numerous methods exist for building these nested complexes [13]. In Section IV we make use of the (*weighted*) *α-complex*, which we now define.

Associate to a point cloud $V = \{v_i\}_{i=1}^n \subset \mathbb{R}^d$ a set of weights $W = \{w_i \geq 0\}_{i=1}^n$ and define the weighted Voronoi cell of the point v_i to be $V(v_i; w_i) := \{x \in \mathbb{R}^d \mid d_{w_i}(v_i, x) \leq d_{w_j}(v_j, x) \text{ for } 1 \leq j \leq n\}$, where $d_{w_i}(v_i, x) = \|v_i - x\|^2 - w_i$. For each scale parameter $\alpha \in \mathbb{R}$, the nonempty closed ball $B_\alpha(x_i; w_i)$ of radius $(\alpha^2 + w_i)^{1/2}$ centered on v_i intersects the weighted Voronoi region to create a convex region, $R_\alpha(v_i; w_i) := B_\alpha(v_i; w_i) \cap V(v_i; w_i)$, containing v_i. The weighted α-complex (at scale α) is then defined to be the collection of subsets of V over which their enclosing convex regions R_α have a nonempty intersection:

$$\mathcal{A}_\alpha(V; W) := \left\{ \sigma \subseteq V \mid \bigcap_{v_i \in \sigma} R_\alpha(v_i; w_i) \neq \emptyset \right\}.$$

Fig. 1 shows several geometric realizations of α-complexes built on a planar point cloud.

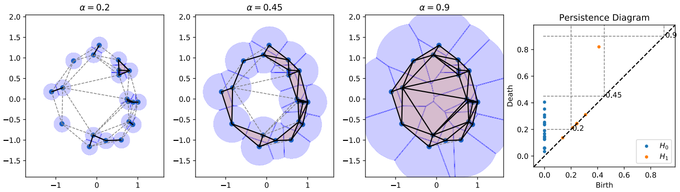

Fig. 1. (Left) Three α-complexes at difference choices of scale parameter α associated to a planar point cloud and (right) the corresponding PD of the filtration over $0 \leq \alpha \leq 1$. Dashed lines connecting points in the cloud indicate the edges in the Delaunay triangulation that have not yet been added to the α-complex. Convex regions, R_α, around each point are drawn in blue. PPs in the quadrant above the dashed line Death = α and to the left of Birth = α correspond to homological features that exist at the scale α.

One benefit of α-complexes is that at large scales they are likely to contain far fewer simplicies than other constructions, since the α-complex is always a subcomplex of the Delaunay triangulation [14] of a point cloud [13]. This can greatly improve the efficiency of computing PH for large point clouds, which is important in many ML settings where it is desirable to have a huge volumne of data samples, or where real-time computations are required.

B. PD Vectorization

PH may be regarded as a collection of transformations, dgm_k, $k \geq 0$ (where k denotes the homological dimension), each of which takes as input a point cloud and produces a multiset of ordered pairs. Each pair in the kth order PD captures the scales at which kth order homology features appear and disappear. It is true that the PH features of a point cloud, as represented in a PD, are discriminating in the sense that the space of PDs, $\mathcal{PD}$, can be endowed with a metric [15]. Thus two point clouds may be compared via the dissimilarity of their PDs. However, the PD representation of homological features as a multiset of ordered pairs is of limited utility to modern ML methods that operate over vector spaces of fixed dimension, since the number of points in a PD may differ significantly between different point clouds, and PPs are not intrinsically ordered. To overcome these limitation and thereby expand the zoo of methods which may take advantage of the structural features captured by PH, numerous methods for 'vectorizing' PDs have been proposed [1], [16]–[25]. Each of these methods are an additional transformation taking a PD to a space endowed with additional structure, and many depend on parameters which must be specified and fixed in order to transform a collection of point clouds into a usable collection of feature vectors that may be input to a learning model. For completeness we define two previously-studied vectorization methods that are used later in Sections IV-A and IV-B.

1) Ranges of Persistence Pairs: Given some data, $\mathcal{X}$, let $D := \text{dgm}_k(\mathcal{X})\{(b_i, d_i)\}_{i=1}^N$ be a kth order PD. A simple way to embed (some of) the content of D into $\mathbb{R}^d$, for some fixed $d \geq 1$, is to first fix an ordering of pairs in D (e.g. decreasing by persistence, $d_i - b_i$) and then concatenate the coordinates of the k birth-death pairs between the jth and the $j + k - 1$st pair in the prescribed ordering; the result is a length-$2k$ vector, PAIRRANGE$(D; j, k)$. Alternatively, one may record only the persistence values, $d_i - b_i$ in the specified range of pairs, giving PERRANGE$(D; j, k)$. Despite it's simplicity, a variant of the latter method revealed in [2] the power of the 0th and 1st order PH feature vectors derived from brain artery trees to explain the age of a subject, for certain choices of j and k.

2) Persistence Images: First transform each PP $(x_i, y_i) \in D$ to birth-persistence coordinates: $(x_i, y_i) \mapsto T(x_i, y_i) := (x_i, y_i - x_i)$. To D associate the surface

$$\rho(D; \sigma, \alpha) := \sum_{u \in T(D)} u_y^\alpha g_u(z; \sigma),$$

where g_u is the 2D isotropic Gaussian $g_u(x, y; \sigma)$,with mean u and variance σ^2. The parameter $\alpha \geq 1$ controls how significantly proximity to the persistence-axis impacts the weight of the Gaussian associated to each PP.

Finally, by discretizing an appropriately chosen rectangular subdomain $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ into mn square 'pixels' of side-length l, the PI of D is defined to be the vector in $\mathbb{R}^{mn}$ gotten by integrating $\rho(D; \sigma, \alpha)$ over each pixel:

$$\text{PERIMAGE}(D; \sigma, \alpha, l)_p := \iint_p \rho(D; \sigma, \alpha)\, dydx,$$

for $p = [x_{\min} + jl, x_{\min} + (j+1)l] \times [y_{\min} + kl, y_{\min} + (k+1)l]$ for each $0 \leq j \leq m - 1$ and $0 \leq k \leq n - 1$.

Summarizing the pipeline defined thus far: data, $\mathcal{X}$, is first transformed into finitely many PDs, $\text{dgm}_k(\mathcal{X})$, $0 \leq k \leq N$, for some choice of N well suited to the data and the problem at hand. Each diagram is then mapped to a finite-dimensional feature vector $\text{VEC}_k(\text{dgm}_k(\mathcal{X}); \theta_k)$, where $\text{VEC}_k : \mathcal{PD} \to \mathbb{R}^{d_k}$ is a parameterized map on the space of PDs, with parameters θ_k. This vectorization is done in anticipation of training a learning model on the resulting vectors. Thus, the topological feature generation step introduces additional dimensions to the CS.

IV. Applications

We applied the TDA-ML hyperparameter optimization pipeline to two biological datasets to explore its effectiveness at simultaneously optimizing configurations that involve both learning model and topological feature generation parameters and to elucidate their interactions in the context of real ML problems.

A. Predicting Synthetic Protein Stability

The authors of [26] aimed to elucidate principles of synthetic protein design by using a software platform Rosetta to design short amino acid (AA) sequences (41-43 AA) meant to stably fold into one of four distinct secondary structure topologies: $\alpha\alpha\alpha$, $\alpha\beta\beta\alpha$, $\beta\alpha\beta\beta$, and $\beta\beta\alpha\beta\beta$, with secondary structures β and α separated by AA loops. We refer to these proteins as the Rocklin designs. The Rosetta-predicted conformations of atomic coordinates in the stable tertiary structure of each design can be used to derive a large collection of biophysical characteristics (e.g. hydrophobicity, total amount of buried nonpolar surface area).

To validate their designs, the authors built and experimentally tested the many thousands of synthetic small proteins for resistance to cleaving by two protease enzymes. Stability scores were then inferred for each protein and for each protease by further modelling the natural resistance of each design to cleaving in its unfolded state. By modelling correlations between the Rosetta-derived biophysical characteristics and the experimental stability scores, the authors of [26] were able to learn features associated with stability, and demonstrate that even linear models trained on model-derived features have some skill at predicting stabilities.

1) Topological Correlates with Protein Stability: The experimental stability scores reported in [26] do not directly measure the strength of inter-atomic interactions, but may reflect the propensity for a protein to remain in a folded conformation that increases resistance to cleaving. Since structural properties of each protein's atomic point cloud may capture how 'tightly packed' a design's stable conformation is, we expect PH features may correlate with stability.

We compute H_0, H_1, and H_2 PDs of the atomic arrangement of each Rocklin design using weighted α-complex filtrations, with weights determined by the van der Waals radius of each atom, and find that, indeed, there are quantitative differences in the topological structures of the most and least stable Rocklin designs. For instance, the least stable designs appear to have fewer, but larger voids, as measured by persistence (Fig. 2). Moreover, even very simple statistical properties of the PDs are found to be correlated with biophysical features, determined by Rosetta's internal models, that are known to be contributors to stability [26] (Fig. 2).

Using random forest classifiers (RFCs), we test the capacity of different topological feature vectors to classify synthetic proteins as stable (experimental stability score greater than 1) or unstable. To define a function to optimize over each CS, we assess model performance using the mean area under the Receiver Operator Characteristic curve [27] (ROCAUC) over 5-fold cross-validation achieved by the ensemble classifier. The classifiers were trained and tested on the topological features of 80/20%-splits of either all available Rocklin designs or on only the subset of $\beta\beta\alpha\beta\beta$ proteins.

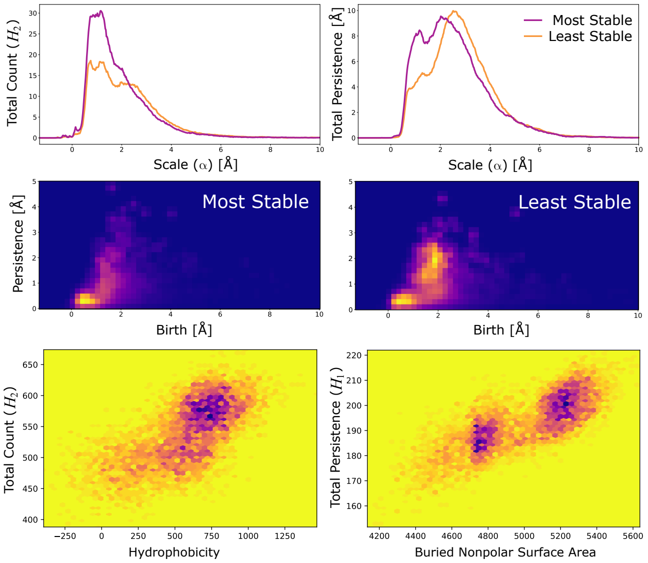

Fig. 2. (Top) The number of (left) and the sum of the persistence (right) of H_2 PPs as a function of the scale parameter α, averaged over the 100 most and least stable Rocklin designs. (Middle) The average of the H_2 PIs of the 100 most (left) and least (right) stable Rocklin designs generated by PERIMAGE$(D; 0.01, 2, 0.2)$ over the region $[-1, 10] \times [0, 5]$ in the birth-persistence plane. (Bottom) binned scatter plots of hydrophobicity versus the number of voids (right) and buried nonpolar surface area versus the sum of persistence of all H_1 pairs for $\beta\beta\alpha\beta\beta$ designs.

2) Bayesian optimization focuses search on promising PD vectorization parameter regions: We first explore the extent to which synthetic protein stability can be predicted using simple H_1 and H_2 topological features that capture loop and void structures in the atomic point clouds of Rosetta-modelled stable conformations. We construct a joint HP space for the PD vectorization methods PAIRRANGE and PERRANGE, represented as a CS,

Configuration Space 1.

```
{
  start pair ($j$): $U(0, 150, 1)$,
  number of pairs ($k$): $U(0, 150, 1)$,
  persistence?: {True, False}
}
```

where $U(0, 150, 1)$ is the uniform distribution over the integers between 0 and 150 and *persistence?* determines if only the persistence values, $d - b$, are concatenated in the feature vector or if both coordinates of each PP, (b, d) are included. In every case, the same configuration is used for both H_1 and H_2 pairs, and the resulting feature vectors are concatenated.

During optimization, a total of 1000 configurations from CS 1 were evaluated. Among the first 20 configurations tested, there are nearly equal numbers with *persistence?* set to True as False, which reflects the uninformed prior on the choice of vectorization methods. As the number of evaluations increases, the better performance of vectorizations which include only the persistence values biases the distribution so the optimization routine spends more time exploring configurations with *persistence?*=True. Nearly 90 percent (880/1000) of all configuration proposals included persistence only values, and the bias becomes clear after only 100 evaluations (Fig. 3). The posterior distribution of *persistence?* reflects the fact that the top ten percent of configurations with *persistence?* $=$ False achieve an average ROCAUC score of only 0.63, while the top ten percent of configurations with *persistence?* $=$ True average 0.77.

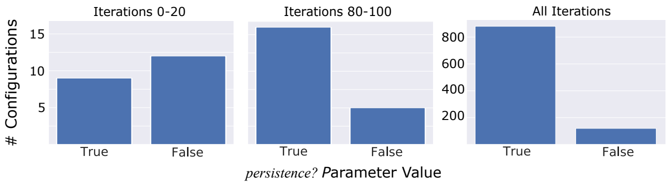

Fig. 3. Empirical distributions of *persistence?* parameter over Bayesian optimization iterations numbers 0-20 (left), 80-100 (middle), and all 1000 configurations.

To test the efficiency of this framework at identifying the global optimal configuration, we performed brute force searches over the slices of the CS with *persistence?* $=$ True, and *persistence?* $=$ False for every even j, k with $0 \leq j < 150$ and $2 \leq k < 150 - j$. The global maximum ROCAUC score was achieved using only persistence values for the PP range 28 to 86, with a ROCAUC score of approximately 0.7738. During optimization the best observed PP range was found to be 25 to 96, with a remarkably similar ROCAUC score of 0.7735. Interestingly, using both birth and death coordinates yields a maximum of only 0.6718 across the range of tested pairs. This drop in performance may be due to an increase in the dimension of the feature space with less informative features, and certainly explains the optimization's bias towards sampling from the subspace *persistence?* $=$ True. A heatmap of the start and end PP indices explored during optimization, together with the best observed configuration and the absolute optimal configuration parameters (shown in Fig. 4), shows agreement between the HP region of highest ROCAUC scores and the region sampled most frequently during optimization.

3) Optimal vectorization parameters may depend on the homological dimension: Here we consider the impact of simultaneously optimizing the learning model HPs, and the PD vectorization HPs in which the ranges of PPs can vary with homological dimension. Using RFCs we predict protein stability across the entire Rocklin dataset by training on a fixed subset of 80 percent of designs randomly selected from all four secondary structure topologies, and testing on the remaining 20 percent of designs. The subspace of the CS controlling the model HPs is chosen to be

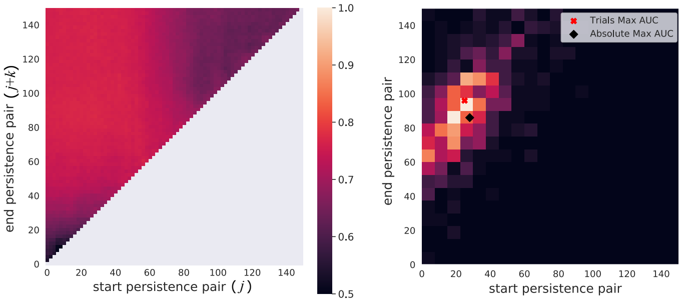

Fig. 4. (Left) Mean 5-fold cross-validation ROCAUC score of a RFC trained on the vectors PERRANGE($\cdot\,; j, k$) derived from Rocklin $\beta\beta\alpha\beta\beta$ designs. (Right) Empirical distribution of PP start and end indices, for pairs ordered by persistence, over 1000 configurations evaluated during Bayesian optimization, with the best performing configuration found during optimization (red x) and the absolute maximum ROCAUC score found by grid search (black diamond).

Configuration Space 2.

{
min. samples per leaf: $U(0, 0.5)$,
min. samples to split: $U(0, 0.5)$,
max. features: $U(0, 1)$
}

and the vectorization configuration subspace includes two copies of CS 1 with *persistence?* =True. See the scikit-learn [28] RFC documentation for a complete description of the available parameters.

Over 325 iterations of Bayesian optimization, we observe a steady improvement in the average performance of the TDA-ML pipeline (Fig. 5 right). Not surprisingly, a major contributor to overall performance is the MSL. For medium to large fractions of the total sample size ($\approx$ $0.3 - 0.5$), performance can be no better than random (ROCAUC = 0.5) due to underfitting.

The number of PPs used to generate each feature vector and the maximum features become more narrowly distributed around the optimum configuration, which yields a ROCAUC of 0.73, towards the end of the optimization. As expected, the histograms of individual HPs in the tail of the iterations of the optimization routine contrasts the promising and the poor-performing regions for each parameter. Fig. 5 illustrates the dramatic difference in the numbers of H_1 and H_2 pairs that yield the best model performance.

4) Learner parameters may influence optimal choices of feature generation parameters: Bayesian optimization of PD vectorization HPs is equally well suited to regression problems as it is to classification problems. Fig. 6 shows the configurations drawn during minimization of the average 5-fold cross-validation root mean squared error (RMSE) between a random forest regressor's (RFR) predictions and the experimentally-derived stability scores, over CS 3, for four different choices of the MSL parameter. Model training and validation was performed on the feature vectors PERIMAGE($\cdot\,; \sigma, 1, l$) of $\beta\beta\alpha\beta\beta$ designs, with σ and l measured in Angstroms.

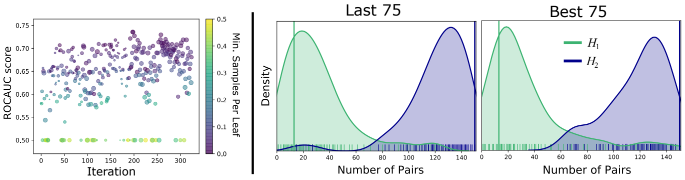

Fig. 5. (Left) ROCAUC scores for each of 325 configurations drawn using Bayesian optimization. Iterations are colored according to the RFC hyperparmaeter determining the minimum number of samples required to define a leaf node in each decision tree (quantified as a fraction of the total sample size), and are scaled by the parameter determining the maximum number of features that are used when splitting each node in a tree during training. (Right) Kernel density estimates of the marginal distributions of HPs determining the number of H_1 and H_2 PP feature vectors, estimated from the final (left) and best (right) 75 iterations of optimization. Vertical lines indicate the best choices of each parameter found over all iterations.

Configuration Space 3.

{
pixel dimensions (l): $U(0.1, 1)$,
kernel bandwidth (σ): $U(0, 0.5)$
}

As MSL increases, the regions of CS 3 yielding the lowest RMSE notably shift towards larger kernel bandwidths and increased pixel dimensions (Fig. 6). This suggests a somewhat counterintuitive relationship between a RFR learning model parameter controlling over/under-fitting and the dimension and fidelity of the PI representations of the PDs.

Over the range of tested MSL values, the empirical optimal RMSE increases with MSL by close to 2%. On the other hand, for a fixed MSL we see optimal performance over CS 3 improve by as much as 5% over the worst configurations sampled. These observations highlight the importance of simultaneous optimization of both model and feature generation HPs.

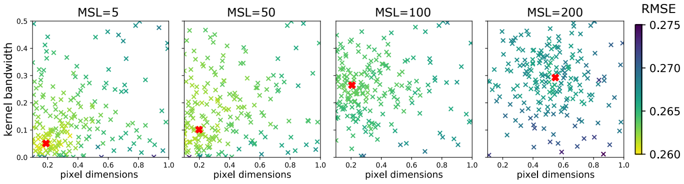

Fig. 6. 200 configurations drawn during optimization iterations over CS 3 with MSL = 5 (top left), 50 (top right), 100 (bottom left), 200 (bottom right). Samples are colored according to the mean RMSE over 5 training/testing splits. Red x's indicate the average HPs of the 20 best performing configurations found during optimization.

B. Predicting Trabecular Number in Porous Bone Tissue

As our second application, we examine 3D point clouds sampled from the human femur. The samples were obtained using a micro-CT scanner concentrated on lighter, more flexible regions known as "trabecular bone tissue" [29]. One commonly accepted statistic used in morphometric analysis of such shapes is the "trabecular number," which is the inverse of the average distance between branches of the medial axis of the shape, reported in units of (1/mm). Due to the porous nature of trabecular bone tissue, some authors have already used topological techniques based on the Euler characteristic at a fixed scale [29] to characterize them, and they have shown a strong correlation between their statistics and trabecular number. Here, we seek to complement their work by learning a mapping from vectorized unweighted α filtrations to the trabecular number.

In this experiment, we take a subset of the data reported in [29] consisting of CT scans of the femoral heads of 6 subjects. The trabecular number varies from 0.57 (1/mm) to 1.13 (1/mm) across subjects. For each subject, we take three disjoint rectangular regions, each of approximately 40mm^3, along the principal stress trajectories in the bone, and we compute H_1 and H_2 PDs of unweighted α filtrations. Each region consists of about 30k vertices, so using α filtrations is crucial to cut down on the number of simplices involved in H_2 in particular. In the PIs, we restrict the birth times to the range [0mm, 0.5mm] and the persistence range between [0mm, 0.6mm].

Once we have the filtrations for each region, we learn a linear map A from the regions to the trabecular numbers of their associated subjects using leave-one-subject-out cross-validation (training on 15 regions and testing on 3 regions for each subject). Since there are far more parameters than data samples in every model, we use Ridge regression as a regularized linear regressor; that is, given a vectorized persistence diagram w, a regularization parameter β, and a trabecular number y, we minimize the objective function $||y - A \cdot w||_2^2 + \beta ||A||_2^2$ over all samples in each training set. We now detail some conclusions from our experiments that shed light on the pipeline.

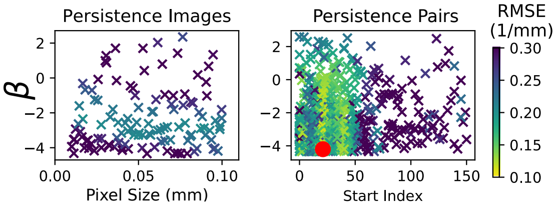

Fig. 7. 1000 configurations drawn during optimization iterations for the bone trabecular number, fixing the kernel width at 0.05 for PIs (left column) and fixing the number of pairs at 50 for persistence pairs (right column). The red dot indicates the optimal configuration found.

1) The optimal vectorization method is problem specific: We first perform an experiment in which we allow the optimization to choose between sorted persistence pairs and persistence images, as well as to vary hyperparameters therein. As Fig. 7 shows that sorted persistences reach about half of the RMSE of the best performing persistence image. As a result, the optimizer ends up sampling configurations much more form the former (right column). A similar conditional configuration space was tested in the protein application, and the exact opposite was true: PIs significantly outperformed the models trained on sorted PPs.

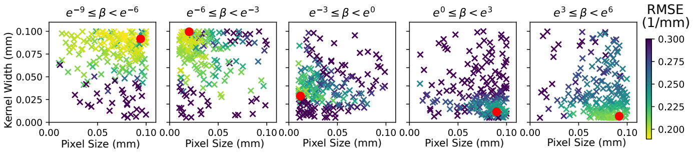

Fig. 8. The result of sampling 200 configurations each for different allowed ranges of the ridge regression regularization parameter β. In the figure above, the ranges increase from left to right, and samples are plotted in the pixel size / kernel bandwidth parameter space of persistence images, with colors indicating RMSE of the corresponding configuration and a red dot drawn over the optimal configuration in each range.

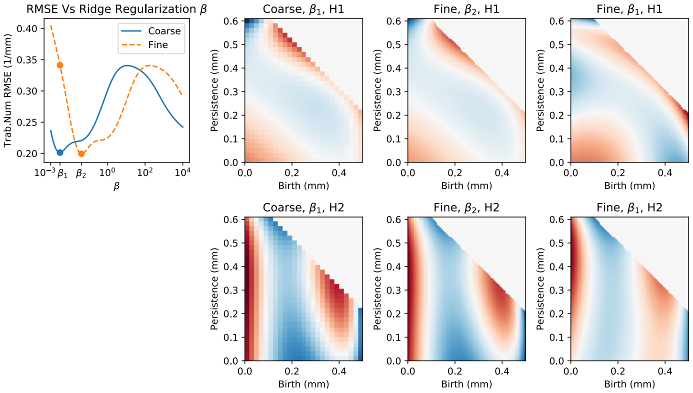

Fig. 9. The result of applying ridge regression fixing $\sigma = 0.05$ and changing l from coarse 0.02mm to a finer 0.005mm resolution. The leftmost plot shows the RMSE versus alpha for the coarse (blue solid) and fine (orange dotted) l. The right three columns show the coefficients of A for both H_1 and H_2 in the top and bottom rows, respectively, for β_1 and β_2 in different cases. Blue values indicate a negative contribution to the trabecular number, while red regions indicate a positive contribution.

2) Kernel and pixel size can act as regularization params: We also observe a notable interaction between the regularization parameter β in ridge regression, the pixel dimension l and kernel bandwidth σ parameters controlling the PIs. Fig. 8 shows that for a smaller β, a larger σ is needed to prevent overfitting. Also, the pixel size can play a similar role for a fixed σ. Fig. 9 shows the error curves varying β for a fixed $\sigma = 0.05$, for two different choices of the pixel size l. For a finer pixel size, the β yielding optimal performance, β_2, is larger than that for the coarser pixel size, β_1, implying more regularization is needed to deal with a larger number of parameters. However, at this optimum, the coefficients in ridge regression appear to be a nearly perfect "superresolution" version of those at a coarser level, with a similar RMSE. By contrast, using the same $\beta = \beta_1$ leads to higher error for the finer pixels, with coefficients which do not match.

3) Model performance may be mostly invariant over a range of pixel dimensions: Fig. 7 (left) reveals the insensitivity of model prediction error over a large range of PI pixel parameters for a narrow range of the regularization parameter β. We speculate that the reason is that for a fixed (relatively small) kernel bandwidth σ, the values values of the subsets of pixels containing the PPs changes very little, even though the dimension of the resulting feature may change significantly. As l decreases, β does appear to trend upward, to compensate for the increased dimension.

V. Discussion

This paper describes a general framework for simultaneously choosing optimal learning-model and topological feature vector HPs. We propose using state-of-the-art Bayesian optimization to efficiently search complex, conditional CSs when computing topological features for use with machine-learning applications.

A detailed exploration of the zoo of PD vectorization methods presently in use is beyond the scope of this work, although many would benefit from the proposed framework. A modular and extensible implementation of this pipeline would enable researchers to systematically compare PD vectorization approaches and eliminate much of the guess work presently needed when choosing representations of PDs.

As shown in Section IV-A, loop and void structures of model-predicted stable conformations are moderately successful at predicting the stability of small synthetic proteins, although this may be largely due to strong correlations with known biophysical determinants of stability. An in-depth analysis comparing and combining biophysical characteristics with topological features is needed to fully understand the importance of homological metrics to protein stability. In this context, it may be highly informative to extract representative subsets of atoms in the equivalence classes of H_1 and H_2 PPs and identify their locations within primary, secondary, and tertiary structures.

In Section IV-B, we find that homological features are well correlated with the trabecular number of porous bone tissue, and that linear models relating H_1 and H_2 features to trabecular number are somewhat generalizable across patients, although the strength of these conclusions would improve with the availability of more data. A more interesting, and difficult problem is to predict other diagnostic measurables such as bone strength, but this is the subject of future work.

One limitation of the proposed framework is that it requires a parameterized configuration space which may be limiting in certain contexts. For example, it is common to choose a parameterized family of weight functions for the PD-to-PI transformation that depends on persistence alone. However, there may be some regions of the PD plane which are more or less discriminating than others, due to similar densities of PPs in those regions across all samples as suggested by Fig. 2 (middle). Parameterizing a family of functions that can reasonably identify and appropriately weight different regions of the plane may be cumbersome, and would not necessarily address the issues of high dimensionality, sparse information content, and spatial correlations in the PI. For these reasons, it is desirable to have a method that could automatically identify the most discriminating regions of the birth-persistence plane given a collection of labelled PDs.

Attributions

This work was supported by the DARPA Synergistic Discovery and Design (SD2) program (Subcontract to FM under

FA8750-17-C-0054), and was made possible through the contributions of many individuals in the SD2 Consortium and beyond. We adopt the CASRAI Contributor Roles Taxonomy (CRediT) for attribution [30]. Contributor roles are defined as (1) Conceptualization, (2) Data curation, (3) Formal analysis, (4) Funding acquisition, (5) Investigation, (6) Methodology, (7) Project administration, (8) Resources, (9) Software, (10) Supervision, (11) Validation, (12) Visualization, (13) Writing – original draft, and (14) Writing – review and editing.

- Francis Motta (1-14), Florida Atlantic University
- Chris Tralie (3, 5, 6, 9, 12-14), Ursinus College
- Marcio Gameiro (2, 9), Rutgers University
- Devin Strickland (2, 8), University of Washington
- Hugh Haddox (2, 8, 9), University of Washington
- Gabe Rocklin (2, 5, 8), Northwestern University
- Hamed Eramian (2, 9), Netrias
- Jed Singer (2, 9), Two Six Labs
- Scott Novotney (2, 9), Two Six Labs
- Nick Leiby (2, 9), Two Six Labs
- Matt Vaughn (8), Texas Advanced Computing Center
- Franco Marinozzi (10),
- Fabiano Bini (10), Università di Roma
- Rosella Bedini (5),
- Gilberto Bini (3), Università degli Studi di Milano
- Steve Haase (4, 10), Duke University
- John Harer (4, 10), Duke University

References

[1] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 218–252, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3122009.3122017

[2] P. Bendich, J. S. Marron, E. Miller, A. Pieloch, and S. Skwerer, "Persistent homology analysis of brain artery trees," *Ann. Appl. Stat.*, vol. 10, no. 1, pp. 198–218, 2016. [Online]. Available: http://dx.doi.org/10.1214/15-AOAS886

[3] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, March 2016, pp. 1310–1315.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[5] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug 2004. [Online]. Available: https://doi.org/10.1023/B:STCO.0000035301.49549.88

[6] M. Feurer and F. Hutter, *Hyperparameter Optimization*. Cham: Springer International Publishing, 2019, pp. 3–33.

[7] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2951–2959.

[8] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2546–2554. [Online]. Available: http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

[9] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, Jan 2016.

[10] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 1. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 115–123. [Online]. Available: http://proceedings.mlr.press/v28/bergstra13.html

[11] F. Hutter, "Automated configuration of algorithms for solving hard computational problems," The University Of British Columbia, Tech. Rep., 2009.

[12] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.

[13] H. Edelsbrunner and J. Harer, *Computational topology: An introduction*. American Mathematical Society, 2010.

[14] B. Delaunay, "Sur la sphere vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, pp. 793–800, 1934.

[15] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[16] D. Rouse, A. Watkins, D. Porter, J. Harer, P. Bendich, N. Strawn, E. Munch, J. DeSena, J. Clarke, J. Gilbert, P. Chin, and A. Newman, "Feature-aided multiple hypothesis tracking using topological and statistical behavior classifiers," in *SPIE Proceedings*, vol. 9474, 2015, p. 94740L.

[17] A. Adcock, E. Carlsson, and G. Carlsson, "The ring of algebraic functions on persistence bar codes," *Homology, Homotopy and Applications*, vol. 18, no. 1, pp. 381–402, 2016.

[18] S. Kališnik, "Tropical coordinates on the space of persistence barcodes," *Foundations of Computational Mathematics*, vol. 19, no. 1, pp. 101–129, Feb 2019. [Online]. Available: https://doi.org/10.1007/s10208-018-9379-y

[19] M. Ferri and C. Landi, "Representing size functions by complex polynomials," *Proc. Math. Met. in Pattern Recognition*, vol. 9, pp. 16–19, 1999.

[20] B. Di Fabio and M. Ferri, "Comparing persistence diagrams through complex vectors," in *International Conference on Image Analysis and Processing 2015 Part I; Editors V. Murino, E. Puppo, LNCS 9279*, 2015, pp. 294–305.

[21] P. Bubenik, "Statistical topological data analysis using persistence landscapes," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 77–102, 2015.

[22] M. Carrière, S. Y. Oudot, and M. Ovsjanikov, "Stable topological signatures for points on 3d shapes," in *Computer Graphics Forum*, vol. 34, no. 5, 2015, pp. 1–12.

[23] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4741–4748.

[24] E. Berry, Y.-C. Chen, J. Cisewski-Kehe, and B. T. Fasy, "Functional Summaries of Persistence Diagrams," *arXiv e-prints*, p. arXiv:1804.01618, Apr 2018, (unpublished).

[25] Q. Zhao and Y. Wang, "Learning metrics for persistence-based summaries and applications for graph classification," *arXiv e-prints*, p. arXiv:1904.12189, Apr 2019, (unpublished).

[26] G. J. Rocklin, T. M. Chidyausiku, I. Goreshnik, A. Ford, S. Houliston, A. Lemak, L. Carter, R. Ravichandran, V. K. Mulligan, A. Chevalier, C. H. Arrowsmith, and D. Baker, "Global analysis of protein folding using massively parallel design, synthesis, and testing," *Science*, vol. 357, no. 6347, pp. 168–175, 2017. [Online]. Available: https://science.sciencemag.org/content/357/6347/168

[27] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recogn.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997. [Online]. Available: http://dx.doi.org/10.1016/S0031-3203(96)00142-2

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] G. Bini, F. Bini, R. Bedini, A. Marinozzi, and F. Marinozzi, "A topological look at human trabecular bone tissue," *Mathematical biosciences*, vol. 288, pp. 159–165, 2017.

[30] A. Brand, L. Allen, M. Altman, M. Hlava, and J. Scott, "Beyond authorship: attribution, contribution, collaboration, and credit," *Learned*

Publishing, vol. 28, no. 2, pp. 151–155, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1087/20150211