Dr. Bei Wang

School of Computing
Scientific Computing and Imaging Institute (SCI)
University of Utah
www.sci.utah.edu/~beiwang
beiwang@sci.utah.edu

March 19, 2019
Deep Learning with Topological Features
The devil is in the detail...
Deep learning and TDA: pipeline

Figure 1: Illustration of the proposed network input layer for topological signatures. Each signature, in the form of a persistence diagram $\mathcal{D} \in \mathbb{D}$ (left), is projected w.r.t. a collection of structure elements. The layer’s learnable parameters $\theta$ are the locations $\mu_i$ and the scales $\sigma_i$ of these elements; $\nu \in \mathbb{R}^+$ is set a-priori and meant to discount the impact of points with low persistence (and, in many cases, of low discriminative power). The layer output $y$ is a concatenation of the projections. In this illustration, $N = 2$ and hence $y = (y_1, y_2)^\top$.

Hofer et al. (2017)
Main idea: transform persistent diagram via an input layer to be used by a neuron network
Computing topological signatures for images

**Figure 2:** Height function filtration of a “clean” (left, green points) and a “noisy” (right, blue points) shape along direction $\mathbf{d} = (0, -1)$. This example demonstrates the insensitivity of homology towards noise, as the added noise only (1) slightly shifts the dominant points (upper left corner) and (2) produces additional points close to the diagonal, which have little impact on the Wasserstein distance and the output of our layer.

Hofer et al. (2017)
Sublevel set filtration of height functions

- Filtration: sub level sets of a height function + essential classes (green)
- Using multiple directions (32 directions)
- Scaling! And \( f \) values are lifted to edges by taking the maximum.
- Extended persistence! (See more on elevation function)

Hofer et al. (2017)
Network architecture

- 32 independent input branches, 1 for each direction
- $i$-th branch gets PDs from directions $d_{i-1}$, $d_i$ and $d_{i+1}$.

https://papers.nips.cc/paper/6761-deep-learning-with-topological-signatures
Computing topological signatures for graphs/networks

Filtration by vertex degree:
\[ f([v_0]) = \text{deg}(v_0) \] (or normalize).
Lift \( f \) to \( K_1 \) by taking the maximum.

Hint: the above pic needs correction!
Using topological signatures is below the state-of-the-art.

The proposed architecture is still better than other approaches that are specifically tailored to the problem.

Most notably, TDA approach does not require any specific data preprocessing, e.g., some sort of contour extraction.
Challenges in using persistence in learning

- Data pre-processing
- Choose filtrations and metrics
- Choose kernels or distance measures
- Choose ML models
- Understand strengths and weaknesses of TDA methods in learning!
Persistence Landscapes
Bubenik (2015); Bubenik and Dlotko (2017)
Persistence landscapes

Bubenik (2015)
Persistence landscapes: implementations

- **Landscape**: implementation of *landscapes*.

Parameters:

<table>
<thead>
<tr>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_landscapes</td>
<td>Number of landscapes.</td>
</tr>
<tr>
<td>resolution</td>
<td>Number of sample points of each landscape.</td>
</tr>
<tr>
<td>ls_range</td>
<td>Range of each landscape. If np.nan, it is set to min and max of x-axis in the diagrams.</td>
</tr>
</tbody>
</table>

- https://github.com/MathieuCarriere/sklearn_tda
- https://scikit-tda.org/libraries.html
- https://github.com/scikit-tda/scikit-tda
- https://github.com/MathieuCarriere/sklearn_tda
Topological Regularizer for Classifiers
Chen et al. (2019)
Figure 1: Comparison of classifiers with different regularizers. For ease of exposition, we only draw training data (blue and orange markers) and the classification boundary (red). (a): our method achieves structural simplicity without over-smoothing the classifier boundary. A standard classifier (e.g., kernel method using the same $\sigma$) could (b) overfit, or (c) overly smooth the classification boundary and reduce overall accuracy. (d): The output of the STOA method based on geometrical simplicity (Bai et al., 2016) also smooths the classifier globally.

Chen et al. (2019)
Measure importance of decision boundaries

Chen et al. (2019)
Some technical details

Given a data set $D = \{(x_n, t_n) \mid n = 1, \ldots, N\}$ and a classifier $f(x, w)$ parameterized by $w$, we define the objective function to optimize as the weighted sum of the per-data loss and our topological penalty.

$$L(f, D) = \sum_{(x, t) \in D} \ell(f(x, w), t) + \lambda L_\mathcal{T}(f(\cdot, w)), \quad (3.1)$$

in which $\lambda$ is the weight of the topological penalty, $L_\mathcal{T}$. And $\ell(f(x, w), t)$ is the standard per-data loss, e.g., cross-entropy loss, quadratic loss or hinge loss.

Chen et al. (2019)

Hinge loss: $\ell(y) = \max(0, 1 - t \cdot y)$, prediction $y$, intended output $t = \pm 1$
Some technical details

\[ L_\mathcal{T}(f) = \sum_{c \in \mathcal{C}(S_f)} \rho(c)^2. \]

**Definition 1 (Robustness).** The robustness of \( c \) is \( \rho(c) = \min_{\hat{f}} \text{dist}(f, \hat{f}) \), so that \( c \) is not a connected component of the boundary of the perturbed function \( \hat{f} \). The distance between \( f \) and its perturbed version \( \hat{f} \) is via the \( L_\infty \) norm, i.e., \( \text{dist}(f, \hat{f}) = \max_{x \in \mathcal{X}} |f(x) - \hat{f}(x)| \).

Chen et al. (2019)
Classic Morse Theory (CMT) and Morse Functions
Edelsbrunner and Harer (2010): B.VI
(Classic) Morse theory studies the topological change of $\mathbb{X}_a$ as $a$ varies.

- $\mathbb{X}$: a compact, smooth $d$-manifold
- $f : \mathbb{X} \to \mathbb{R}$: differentiable
- sublevel set: $\mathbb{X}_a = f^{-1}(-\infty, a]$
- A point $x \in \mathbb{X}$ is critical if the derivative at $x$ equals zero
- $\lambda(x)$: the Morse index of a non-degenerate critical point $x$ is the number of negative eigenvalues in the Hessian matrix
- Next page: $p_1, p_2, p_3, p_4$, index 0, 1, 1, and 2
- $f$ is a Morse function if all critical points are non-degenerate and its values at the critical points are distinct
Example

Goresky and MacPherson (1988)
Two fundamental results of CMT

**Theorem (CMT-A)**

Let $f : X \to \mathbb{R}$ be a differentiable function on a compact smooth manifold $X$.

Let $a < b$ be real values such that $f^{-1}[a, b]$ is compact and contains no critical points of $f$.

Then $X_a$ is diffeomorphic to $X_b$.

- A *diffeomorphism* is an isomorphism of smooth manifolds.
- It is an invertible function that maps one differentiable manifold to another such that both the function and its inverse are smooth.
Two fundamental results of CMT

**Theorem (CMT-B)**

Let \( f \) be a Morse function on \( \mathbb{X} \).

Consider two regular values \( a < b \) such that \( f^{-1}[a, b] \) is compact but contains one critical point \( u \) of \( f \), with index \( \lambda \).

Then \( \mathbb{X}_b \) is homotopy equivalent (diffeomorphic) to the space \( \mathbb{X}_a \cup_B A \), that is, by attaching \( A \) along \( B \).

The pair of spaces \( (A, B) = (D^\lambda \times D^{d-\lambda}, \partial D^\lambda \times D^{d-\lambda}) \) is the Morse data, where \( d \) is the dimension of \( \mathbb{X} \) and \( \lambda \) is the Morse index of \( u \), \( D^k \) denotes the closed \( k \)-dimensional disk and \( \partial D^k \) is its boundary.
Goresky and MacPherson (1988)
\[(A, B) = (D^\lambda \times D^{d-\lambda}, (\partial D^\lambda) \times D^{d-\lambda})\]

<table>
<thead>
<tr>
<th>Critical point</th>
<th>Morse data ((A, B))</th>
</tr>
</thead>
</table>
| \(p_1\)      | \(\left(\begin{array}{c} \text{\ding{55}} \\
\end{array}\right) = (D^0 \times D^2, \partial D^0 \times D^2)\) |
| \(p_2\) or \(p_3\) | \(\left(\begin{array}{c} \text{\ding{55}} \\
\end{array}\right) = (D^1 \times D^1, \partial D^1 \times D^1)\) |
| \(p_4\)      | \(\left(\begin{array}{c} \text{\ding{55}} \\
\end{array}\right) = (D^2 \times D^0, \partial D^2 \times D^0)\) |

Goresky and MacPherson (1988)


