Applying Persistent Homology to the Patch Camelyon Dataset

David Brodsky

MATH 4v91 Class Project

About the Dataset

Patch Camelyon (PCam) dataset

- Contains 327,680 images from histopathologic scans of lymph node sections
- Images are 96 x 96 pixels and RGB
- Positive images contain tumor tissue
- Negative images do not contain tumor tissue
- Dataset is split into train, test, and validation splits

Goal: classify an image as metastatic or non-metastatic

Source: https://github.com/basveeling/pcam

Examples from the Dataset

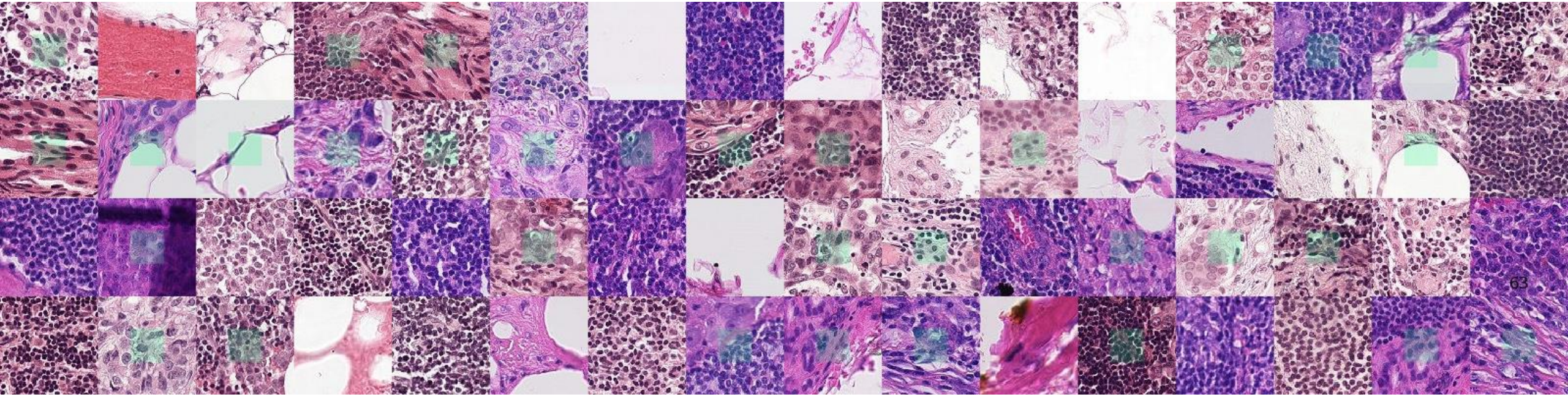

Figure 1: Sample images from the PCam dataset.
Source: https://github.com/basveeling/pcam

Motivations

- Medical imaging is a major application of data science
- Automated metastatic cancer detection would save lives
- PCam is a challenging dataset in this domain

Motivating Questions

- Do TDA methods from this class work well on PCam?
  - No existing papers apply TDA to PCam
- Can TDA be successfully applied to similar datasets?

Methodology (1/2)

1. **Create dataset**
   a. Randomly select images from the “test” split of PCam
   b. Balanced setting: 300 positives, 300 negatives
   c. Imbalanced setting: 300 positives, 700 negatives
2. **Preprocessing**
   a. Convert images to grayscale
   b. Get persistence diagrams (using sublevel or superlevel filtration)
   c. Extract Betti-0 and Betti-1 vectors (200 bins each)
3. **Select features**
   a. Use only Betti-0 vectors
   b. Use Betti-0 and Betti-1 vectors concatenated

Methodology (2/2)

4. **Split dataset**
   a. Use an 80:20 train-test split
   b. Fix the random state so results are reproducible
5. **Perform PCA on training set (OPTIONAL)**
   a. Reduce 200/400 features to 25 features
6. **Train machine learning model**
   a. Train a random forest or XGBoost model on the training set
   b. Tune model hyperparameters
   c. Fix the random seed for reproducibility
   d. Evaluate models on the test set
7. **Repeat training with modified filtration, features, and/or PCA**

Balanced Setting: Sublevel Filtration

METHOD	Train Accuracy	Train AUC	Test Accuracy	Test AUC
PCA + XGBoost (Betti-0)	0.998	0.998	0.725	0.723
XGBoost (Betti-0 & Betti-1)	0.958	0.958	0.708	0.706
PCA + RF (Betti-0)	0.963	0.962	0.700	0.698
RF (Betti-0 & Betti-1)	0.956	0.956	0.700	0.697
XGBoost (Betti-0)	0.994	0.994	0.683	0.681
PCA + RF (Betti-0 & Betti-1)	0.933	0.933	0.667	0.662
RF (Betti-0)	0.988	0.987	0.650	0.647
PCA + XGBoost (Betti-0 & Betti-1)	0.969	0.969	0.642	0.638

Table 1: Accuracy and AUC results using sublevel filtration on the balanced dataset. Results are sorted by test accuracy.

Balanced Setting: Superlevel Filtration

METHOD	Train Accuracy	Train AUC	Test Accuracy	Test AUC
PCA + XGBoost (Betti-0 & Betti-1)	0.963	0.962	0.750	0.749
XGBoost (Betti-0)	0.996	0.996	0.717	0.715
XGBoost (Betti-0 & Betti-1)	1.000	1.000	0.708	0.706
PCA + RF (Betti-0 & Betti-1)	0.994	0.994	0.700	0.696
RF (Betti-0 & Betti-1)	0.977	0.977	0.683	0.680
RF (Betti-0)	0.973	0.973	0.667	0.665
PCA + RF (Betti-0)	0.958	0.958	0.667	0.665
PCA + XGBoost (Betti-0)	0.998	0.998	0.650	0.648

Table 2: Accuracy and AUC results using superlevel filtration on the balanced dataset. Results are sorted by test accuracy.

Imbalanced Case: Sublevel Filtration

METHOD	Train Accuracy	Train AUC	Test Accuracy	Test AUC
RF (Betti-0 & Betti-1)	0.931	0.948	0.880	0.869
RF (Betti-0)	0.969	0.978	0.900	0.867
PCA + RF (Betti-0)	0.964	0.974	0.890	0.865
PCA + XGBoost (Betti-0)	0.931	0.948	0.865	0.863
XGBoost (Betti-0)	0.921	0.938	0.850	0.853
XGBoost (Betti-0 & Betti-1)	0.929	0.942	0.860	0.844
PCA + XGBoost (Betti-0 & Betti-1)	0.991	0.994	0.875	0.839
PCA + RF (Betti-0 & Betti-1)	0.953	0.958	0.870	0.830

Table 3: Accuracy and AUC results using sublevel filtration on the imbalanced case. Results are sorted by test AUC.

Imbalanced Case: Superlevel Filtration

METHOD	Train Accuracy	Train AUC	Test Accuracy	Test AUC
XGBoost (Betti-0 & Betti-1)	0.945	0.958	0.890	0.870
RF (Betti-0 & Betti-1)	0.869	0.877	0.870	0.846
PCA + XGBoost (Betti-0 & Betti-1)	0.988	0.991	0.860	0.834
PCA + RF (Betti-0 & Betti-1)	0.976	0.983	0.860	0.823
RF (Betti-0)	0.917	0.925	0.835	0.805
PCA + XGBoost (Betti-0)	0.935	0.946	0.825	0.804
PCA + RF (Betti-0)	0.906	0.914	0.825	0.798
XGBoost (Betti-0)	0.904	0.915	0.820	0.790

Table 4: Accuracy and AUC results using superlevel filtration on the balanced case. Results are sorted by test AUC.

Comparison to Existing Results

Model	Paper	Year	AUC
DSF-CNN (C8)	Dense Steerable Filter CNNs for Exploiting Rotational Symmetry in Histology Images	2020	0.975
Steerable G-CNN (C8)	Learning Steerable Filters for Rotation Equivariant CNNs	2017	0.971
Steerable G-CNN (C8)	Learning Steerable Filters for Rotation Equivariant CNNs	2017	0.969
Steerable G-CNN (C12)	Learning Steerable Filters for Rotation Equivariant CNNs	2017	0.969
G-CNN (C8)	Roto-Translation Equivariant Convolutional Networks: Application to Histopathology Image Analysis	2020	0.968
TDA (imbalanced setting)	This presentation	2022	0.870
TDA (balanced setting)	This presentation	2022	0.749

Table 5: Comparison to top existing results.
Source: https://paperswithcode.com/sota/breast-tumour-classification-on-pcam

Analysis of Results

Our Results

- Our best results were
  - 75.0% test accuracy (balanced setting)
  - 87.0% test AUC (imbalanced setting)
- Multiple methods achieve comparable results
- Results are respectable given the size of the training dataset

Areas for Improvement

- Train on a larger dataset
- Use other TDA tools
- Use other dimension reduction techniques

Questions?

Thank You!