Topologically Faithful Image Segmentation via Induced Matching of Persistence Barcodes

Topological Data Analaysis - Student seminar

Gauthier Multari, Simon Queric and Vincent Herfeld

ENS Paris-Saclay

December 12, 2023

Table of content

How to use topology for segmentation ?

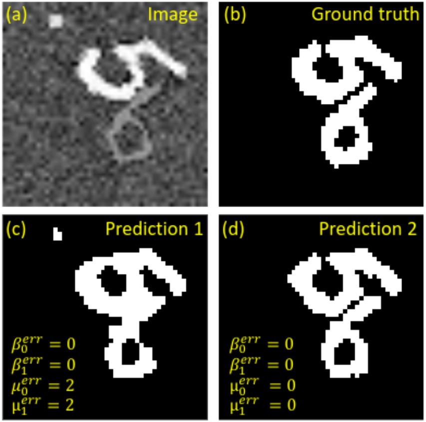


Figure 1: Exemplary segmentations of identical Dice scores from models trained with *Wasserstein loss* (c) and Betti matching loss (d). Dice and Betti number error (β^{err}) are indecisive between both predictions. On the other hand, our Betti matching error (μ^{err}) favors the superior segmentation in (d).

1. Classical metrics for image segmentation (Dice loss, binary-cross entropy) don't take into account topology features
2. First idea that came in mind : the Betti number error.

$$d_{\text{Betti}}(X, Y) = \sum_{d=0}^{+\infty} |\beta_d(X) - \beta_d(Y)|$$

Obvious disadvantage of the betti number error : not a local measure, doesn't take into acount spatially related features. Poor metric for segmentation accuracy.

Wasserstein loss

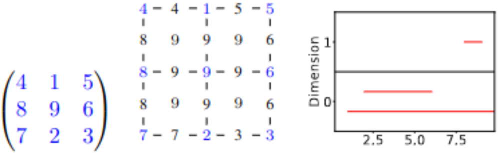
- In practice, people used the Wasserstein loss to train NN (related to Betti number error, more or less the same)
- $\ell_W(L, G) = \min_{\gamma:\mathsf{Dgm}(L)\to\mathsf{Dgm}(G)} \sum_{q\in\mathsf{Dgm}(L)} \|q - \gamma(q)\|_2^2$
- Wasserstein loss doesn't guarantee that matched points represent spatially related features.
- It's not sensitive to the location of topological features

Table of content

Images as filtered cubical complexes

(a) K_1 (b) K_2 (c) K_3

Figure 3: A filtered cubical complex with varying homology in degree 1. Adding the green 1-cell in (b) creates homology (birth) and adding the red 2-cell in (c) turns homology trivial (death). Together they form a *persistence pair*.



(a) $\boldsymbol{I}$ (b) $f_{\boldsymbol{I}}: K^{3,3} \to \mathbb{R}$ (c) $\mathcal{B}(I)$

Table of content

Induced matching between barcodes

Theorem 2.1 Let $\phi : M \to N$ be a morphism of p.d.f, staggered persistence modules that are continuous from above. Then there are unique injective maps $\mathcal{B}(\text{im}\phi) \hookrightarrow \mathcal{B}(M)$ and $\mathcal{B}(\text{im}\phi) \hookrightarrow \mathcal{B}(N)$, which map an interval $[b, c) \in \mathcal{B}(\text{im}\phi)$ to an interval $[b, d) \in \mathcal{B}(M)$ wih $c \leq d$ and to an interval $[a, c) \in \mathcal{B}(N)$ with $a \leq b$, respectively.

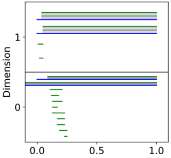
- Differences between induced matching and Wasserstein matching.
- *Induced Matchings and the Algebraic Stability of Persistence Barcodes* by Ulrich Bauerand and Michael Lesnick, 2015

Algorithm

Procedure InducedMatching$(\mathcal{B}(\boldsymbol{I},\boldsymbol{J}), \mathcal{B}(\boldsymbol{I}), \mathcal{B}(\boldsymbol{J}))$
    $\sigma(\boldsymbol{I},\boldsymbol{J})_0, \sigma(\boldsymbol{I},\boldsymbol{J})_1 = \phi$; // Initialize matched refined intervals
    for $d \leftarrow 0$ **to** 1 **by** 1 **do** // Loop over dimension d
        foreach $(a,b) \in \mathcal{B}(\boldsymbol{I},\boldsymbol{J})_d$ **do** // For each image persistence pair
            $m_i, m_j = None$;
            foreach $(c,d) \in \mathcal{B}(\boldsymbol{I})_d$ **do** // Match left endpoints
                if $c = a$ **then**
                    $m_i = (c,d)$;
                    break
                end
            end
            if $m_i = None$ **then** // Skip search if no match found
                continue
            end
            foreach $(c,d) \in \mathcal{B}(\boldsymbol{J})_d$ **do** // Match right endpoints
                if $d = b$ **then**
                    $m_j = (c,d)$;
                    break
                end
            end
            if $m_j = None$ **then** // Skip search if no match found
                continue
            end
            Add $(m_i, (a,b), m_j)$ to $\sigma(\boldsymbol{I},\boldsymbol{J})_d$;
        end
    end
    return $(\sigma(\boldsymbol{I},\boldsymbol{J})_0, \sigma(\boldsymbol{I},\boldsymbol{J})_1)$

(c) $\boldsymbol{C}$

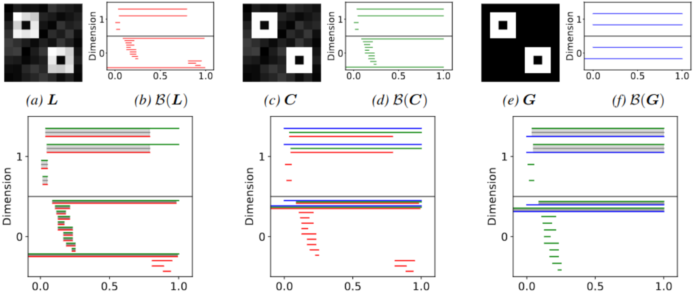
(e) $\boldsymbol{G}$



(i) $\sigma(\boldsymbol{G},\boldsymbol{C})$

Betti matching

- Betti matching is a spatially renforced Betti error.
- **Best idea of the paper** : introduction of a comparison image between the likelihood map L and the ground truth G.
- **Definition** $C = \max(L, G)$ (element-wise maximum).
- **Definition** (Betti matching) $\mu(L, G) = \sigma(G, C)^{-1} \circ \sigma(L, C)$

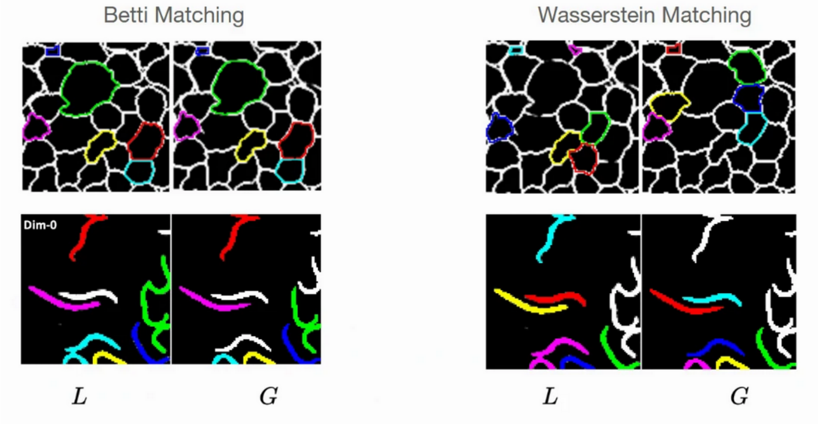


(a) $\boldsymbol{L}$ *(b)* $\mathcal{B}(\boldsymbol{L})$ *(c)* $\boldsymbol{C}$ *(d)* $\mathcal{B}(\boldsymbol{C})$ *(e)* $\boldsymbol{G}$ *(f)* $\mathcal{B}(\boldsymbol{G})$

(g) $\sigma(\boldsymbol{L}, \boldsymbol{C})$ *(h)* $\mu(\boldsymbol{L}, \boldsymbol{G})$ *(i)* $\sigma(\boldsymbol{G}, \boldsymbol{C})$

Figure 6: **Betti matching.** (a)–(f) show a likelihood map $\boldsymbol{L}$, a ground truth $\boldsymbol{G}$, the comparison image $\boldsymbol{C}$ and their barcodes. (g) and (i) show the induced matchings $\sigma(\boldsymbol{L}, \boldsymbol{C}) \colon \mathcal{B}(\boldsymbol{L}) \to \mathcal{B}(\boldsymbol{C})$ and $\sigma(\boldsymbol{G}, \boldsymbol{C}) \colon \mathcal{B}(\boldsymbol{G}) \to \mathcal{B}(\boldsymbol{C})$ (matchings indicated in gray) and (h) shows the resulting Betti matching $\mu(\boldsymbol{L}, \boldsymbol{G}) \colon \mathcal{B}(\boldsymbol{L}) \to \mathcal{B}(\boldsymbol{G})$, which matches a red interval to a blue interval if there is a green interval in between. We use this matching to define our loss and metric.

Spatially correct matching of features



Betti matching loss

- **Definition** $\ell_{\mathsf{BM}}(L,G) = \sum_{q \in \mathsf{Dgm}(L)} 2\|q - \mu(L,G)(q)\|_2^2$
- Let **P** the binarized prediction of **L**. Then: $\ell_{BM}(P,G) = \#ker(\mu(P,G)) + \#coker(\mu(P,G)) =: \mu^{err}(P,G)$
- In the neural network : $\ell_{\mathsf{train}}(L,G) = \alpha\ell_{\mathsf{BM}}(L,G) + \ell_{\mathsf{dice}}(L,G)$ with $\alpha > 0$.

Table of content

Recap of the pipeline

1. Computation of barcodes for L, C and G with respect to the superlevel set filtration.
2. Induced matchings $\sigma(L, C), \sigma(G, C)$.
3. Betti matching error: composition of induced matchings.
4. Betti matching loss.

Results

	Loss	Dice ↑	clDice ↑	Acc. ↑	μ^{err} ↓	μ_0^{err} ↓	μ_1^{err} ↓	β^{err} ↓	β_0^{err} ↓	β_1^{err} ↓
CREMI	Dice	0.894	0.939	0.959	149.64	39.68	109.96	114.12	39.12	75.00
	clDice	0.879	**0.944**	0.952	147.04	34.36	112.68	103.92	33.64	70.28
	Hu et al.	0.888	0.935	0.957	162.48	44.24	118.24	118.16	43.68	74.48
	Ours	0.893	0.941	0.959	**129.80**	**31.00**	**98.80**	**79.16**	**30.36**	**48.80**
Roads	Dice	0.663	0.698	0.974	117.80	87.04	30.76	113.96	86.54	27.42
	clDice	0.668	0.704	0.975	131.00	102.08	28.92	125.83	101.67	24.17
	Hu et al.	0.674	0.712	0.974	101.00	73.04	27.96	95.83	72.54	23.29
	Ours	0.663	0.713	0.972	**83.00**	**56.30**	26.70	**75.08**	**55.79**	**19.29**
synMnist	Dice	0.871	0.907	0.962	3.70	1.96	1.74	2.590	1.674	0.916
	clDice	0.875	0.921	0.963	2.54	0.87	1.67	1.640	0.700	0.940
	Hu et al.	0.866	0.915	0.960	2.85	1.00	1.85	1.802	0.764	1.038
	Ours	0.849	0.915	0.954	**2.28**	**0.53**	1.75	**1.348**	**0.426**	0.922
Elegans	Dice	0.922	0.959	0.984	4.10	2.60	1.50	2.60	1.40	1.20
	clDice	0.917	**0.964**	0.982	3.90	2.20	1.70	2.20	1.20	1.00
	Hu et al.	0.921	0.959	0.984	4.30	2.84	1.45	2.50	1.35	1.15
	Ours	0.919	0.960	0.983	**3.40**	2.10	**1.30**	**1.90**	**0.80**	1.10
Colon	Dice	0.899	0.863	0.970	44.26	21.76	22.50	33.75	13.75	20.00
	clDice	0.907	0.871	0.974	47.26	18.76	28.50	37.75	11.75	26.00
	Hu et al.	0.902	0.876	0.972	34.50	15.50	19.00	22.00	7.00	15.00
	Ours	0.907	0.871	0.975	**32.00**	**14.26**	17.76	21.50	**6.25**	15.25
Buildings	Dice	0.623	0.672	0.934	572.44	551.00	21.46	162.95	151.70	11.25
	clDice	0.632	**0.693**	0.931	571.20	535.96	35.26	175.50	155.05	20.45
	Hu et al.	0.625	0.677	0.934	556.60	537.50	19.10	181.10	169.60	11.50
	Ours	0.625	0.685	**0.937**	**489.16**	**471.26**	17.90	**118.45**	**107.75**	10.70

Discussion

- **Strengths**
  1. Improve topological accuracy
  2. Ensures proper feature matching during training
  3. Reproducibility : code available on github, PyTorch implementation
- **Weaknesses**
  1. Complexity : $O(n^3)$ with $n = NM$ for images of size (N, M).

Table of content

Ulrich Bauer

Dear Mr. Queric, The spatial information is what is used to align the images. With this, it becomes possible to construct the comparison image, and to obtain the inclusion maps. Without these maps, it would not be possible to obtain induced matchings; they are always induced by maps. The Wasserstein matching does not use any of this, and only looks at the persistence diagrams. As a result, it can't tell whether two features correspond in some way or not. The alignment and the inclusion maps are our way of constructing such a correspondence.

References

1. Topologically Faithful Image Segmentation via Induced Matching of Persistence Barcodes, Stucki et al., 2023
2. Induced Matchings and the Algebraic Stability of Persistence Barcodes, Ulrich Bauer and Michael Lesnick, 2015