A Tutorial on Topological Data Analysis in Text Mining

Shafie Gholizadeh [1] Wlodek Zadrozny [1 2]

[1]College of Computing, University of North Carolina at Charlotte

[2]School of Data Science, University of North Carolina at Charlotte

Agenda

#	Segment	Presentation	Time
1	Introduction	Slides	15 min
2	TDA background & available software	Slides/Code	20 min
3	Positioning TDA in Text Mining	Slides	15 min
4	TDA methods for Text Mining	Slides/Code	40 min
5	Limitations, opportunities & conclusion	Slides	15 min

Contents

Introduction: Searching for New Representations of Text

How do we classify, cluster, or generally analyze text?

- Term Frequency space
- Embeddings
- NNs

Some basic questions:

How do we classify long text?

Why should we limit ourselves to the conventional features?

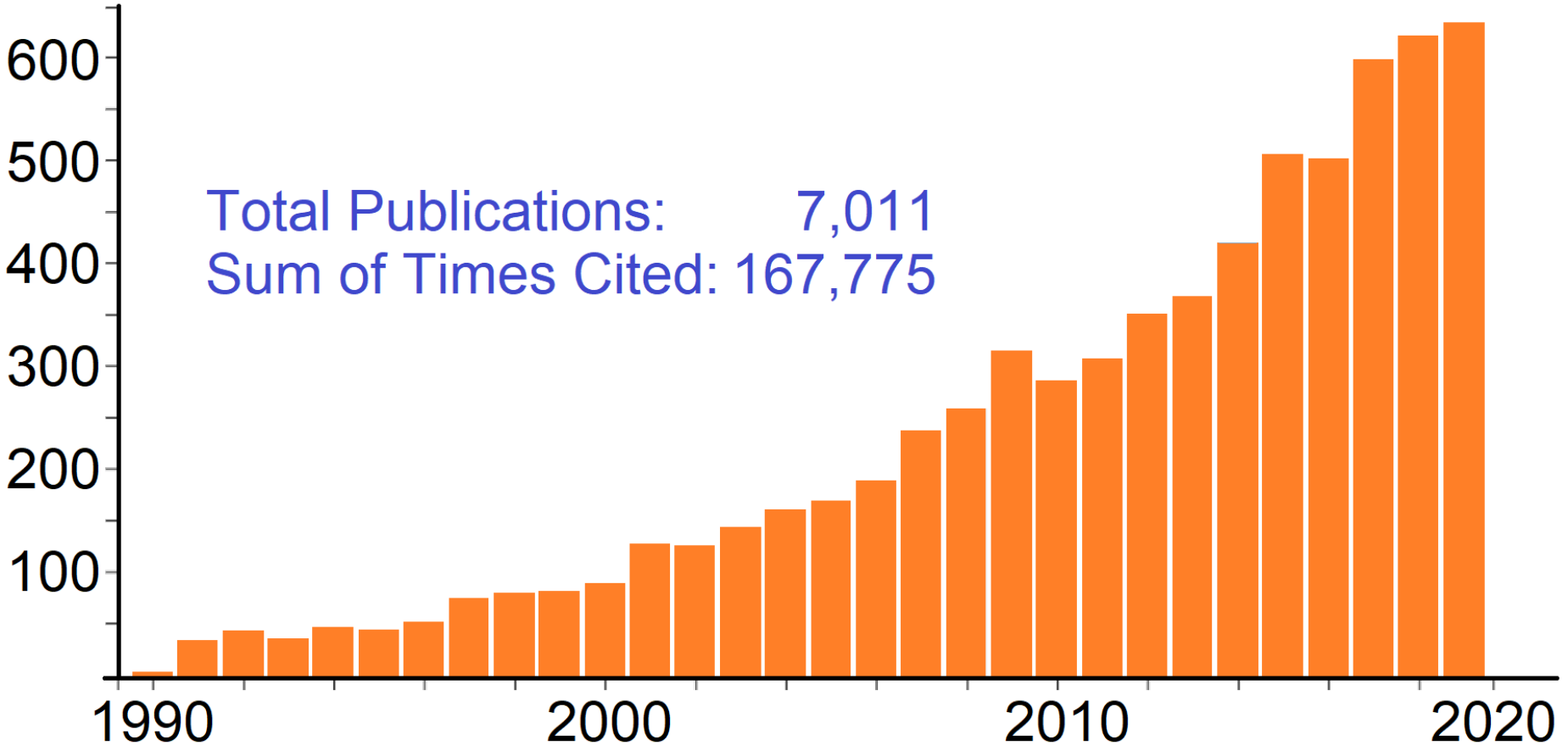
Is there any alternative to NNs?

This tutorial targets new text representations using topological data analysis.

Introduction: Topological Data Analysis (TDA)

TDA is becoming more popular as a research area.

Publications on TDA[1]

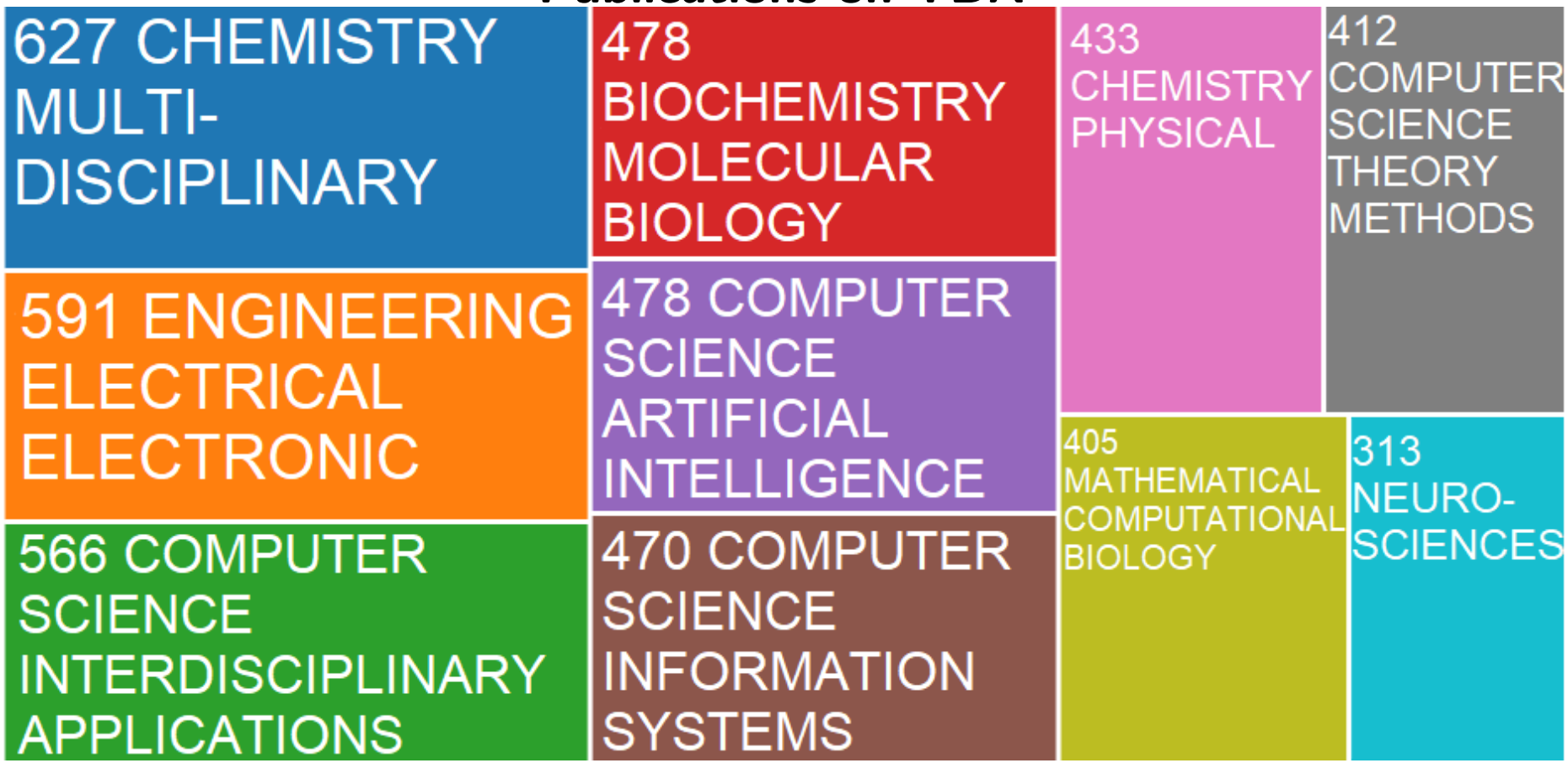


[1] Web of Science portal, retrieved on April 20, 2020

Introduction: Topological Data Analysis (TDA)

TDA is becoming more popular as a research area.

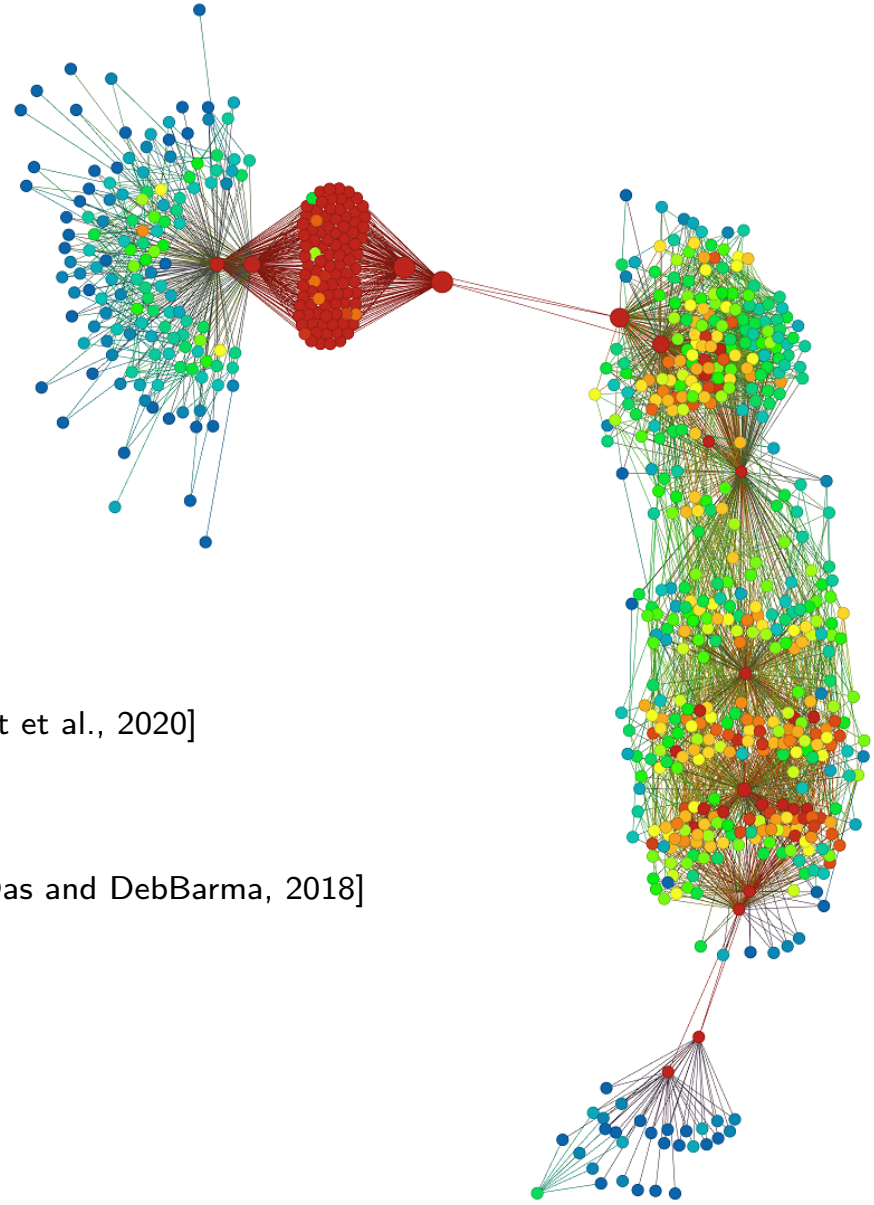
Publications on TDA[2]



[2] Web of Science portal, retrieved on April 20, 2020

Some Recent Contributions of TDA

- Clustering
- Dimensionality Reduction
- Descriptive modeling



- Neural Nets Architecture

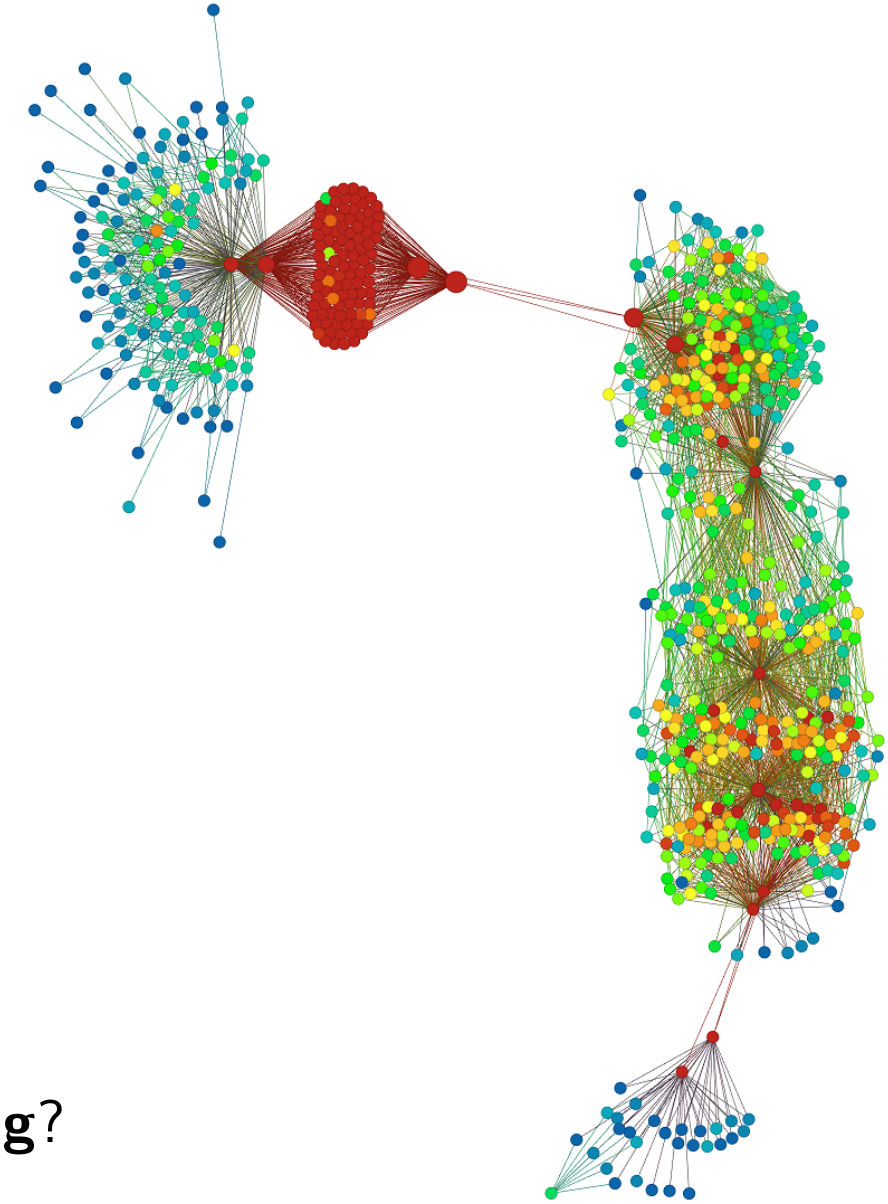
  [Guss and Salakhutdinov, 2018] [Hofer et al., 2019] [Naitzat et al., 2020]

- Sensor Network Coverage

  [De Silva and Ghrist, 2006] [Adams and Carlsson, 2015], [Das and DebBarma, 2018]

- Time Series Analysis
- Signal Processing
- Dynamical Systems Analysis

Introduction: Topological Data Analysis (TDA)

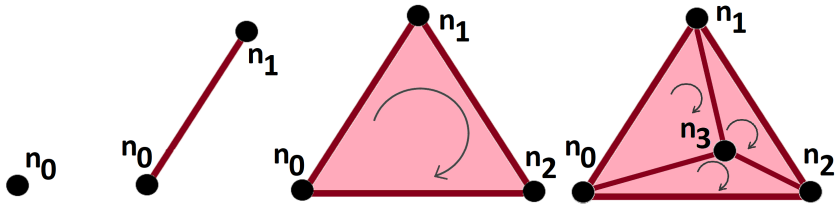


How to find **shapes** in text?

How to use **TDA** in **text processing**?

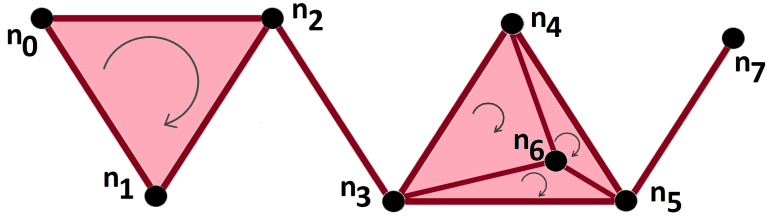
What Is TDA?

- TDA: A collection of methods that find structure of shapes in data.



- What's the Ancestry of TDA?

  Computational Geometry $\Rightarrow$
  Computational Topology $\Rightarrow$
  Topological Data Analysis ■



- Common Approach in TDA is to:

  (1) Capture the shapes as the main characteristics.
  (2) Dismiss the rest as noise or irrelevant information.

TDA in Text Mining: How?

- Why Important?
  - Use Topological features in addition to the other features.
  - Capture the order in the text.
- Borrowing Ideas from Time-Series Analysis
  - Consider text as n-dimensional time-series of n entities.
- Designing Order-Preserving Text Processing

Our Contributions

- Providing a New Framework and Algorithms to Extract "Topological Features" from Text:
  - Extracting TDA features from word embeddings space
  - Extracting TDA features from TF-IDF space
  - Extracting TDA features without using conventional representations
- Showing the value of TDA in Text Mining
  - TDA features carrying exclusive information that is not reflected in conventional features.

Contents

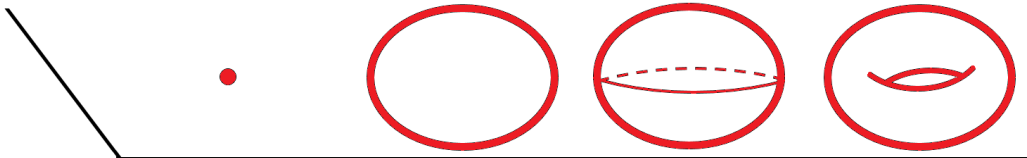
Betti Numbers Capture Topological Structure.

- The i^{th} Betti number: number of i-dimensional holes a in a shape.

β_0: Number of connected components

β_1: Number of 1-D holes

β_2: Number of 2-D voids



β_0	1	1	1	1
β_1	0	1	0	2
β_2	0	0	1	1
β_3	0	0	0	0
⋮	⋮	⋮	⋮	⋮

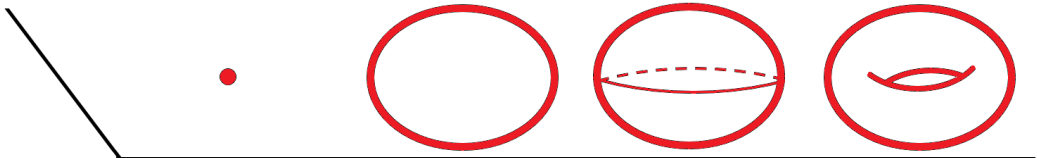
Betti Numbers Capture Topological Structure.

- The i^{th} Betti number: number of i-dimensional holes a in a shape.

β_0: Number of connected components

β_1: Number of 1-D holes

β_2: Number of 2-D voids



β_0	1	1	1	1
β_1	0	1	0	2
β_2	0	0	1	1
β_3	0	0	0	0
⋮	⋮	⋮	⋮	⋮

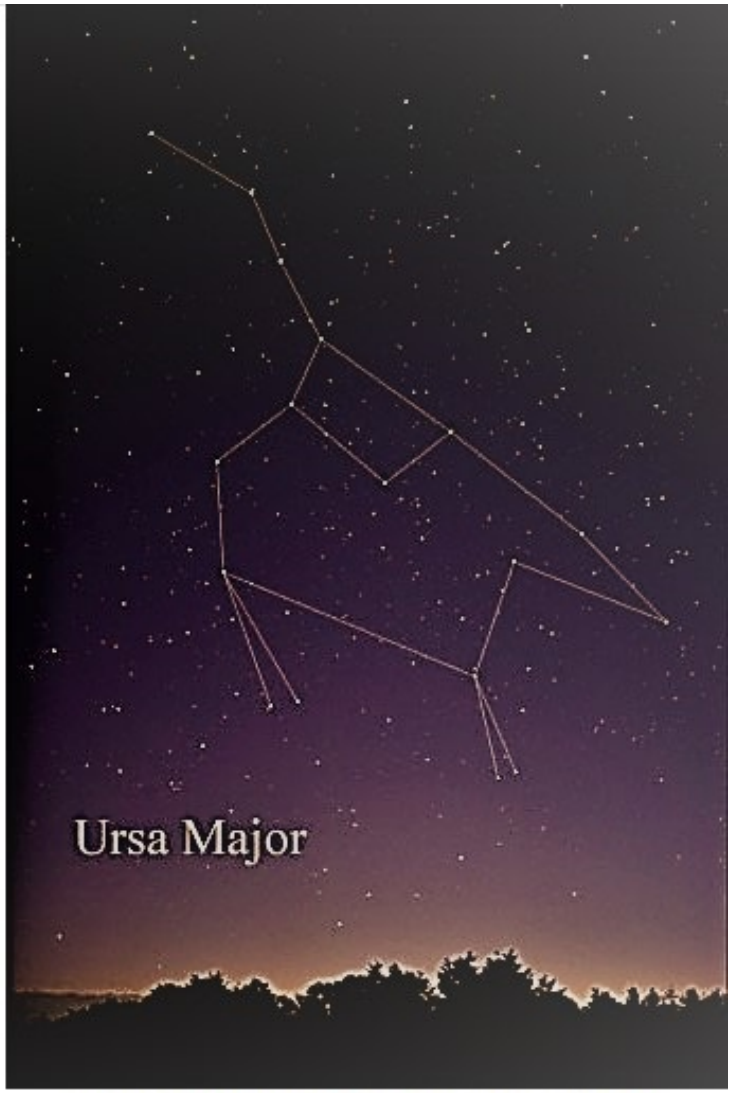
- βs are robust under stretching or shrinking.
- βs simplify complex information.
- Homology studies βs.
- To find the structure of shapes:
  - Capture the shapes as the main characteristics.
  - Dismiss the rest as noise or irrelevant information.

What Is Persistent Homology?

- High dimensional data sets are:
  "huge number of discrete points"
  $\Rightarrow$ There are no continuous shapes!
  $\Rightarrow$ How to define/compute β's?

What Is Persistent Homology?

- High dimensional data sets are:
  "huge number of discrete points"
  $\Rightarrow$ There are no continuous shapes!
  $\Rightarrow$ How to define/compute β's?
- How visual interpretation works?
  infer a continuous shape
  from discrete points.
- Translate points into:
  a meaningful topological structure.

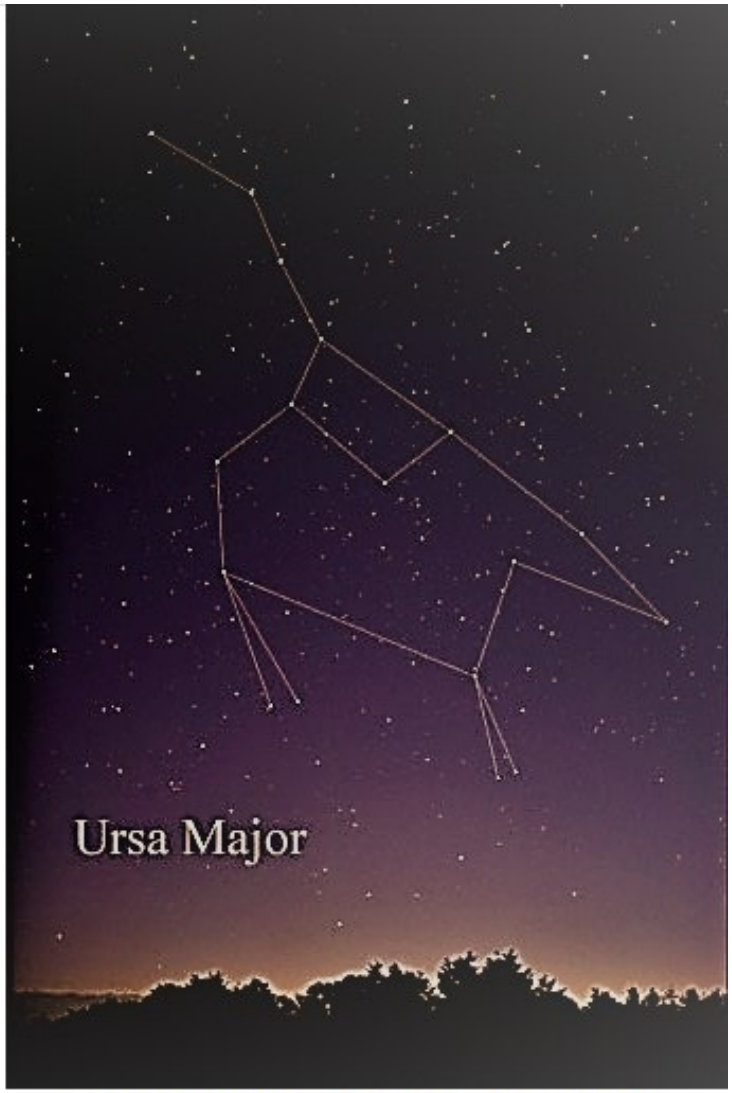


https://en.wikipedia.org/wiki/Ursa_Major

[3] [Edelsbrunner et al., 2000, Zomorodian and Carlsson, 2005]

What Is Persistent Homology?

- High dimensional data sets are:
  "huge number of discrete points"
  $\Rightarrow$ There are no continuous shapes!
  $\Rightarrow$ How to define/compute β's?
- How visual interpretation works?
  infer a continuous shape
  from discrete points.
- Translate points into:
  a meaningful topological structure.
- We need Persistent Homology[3].

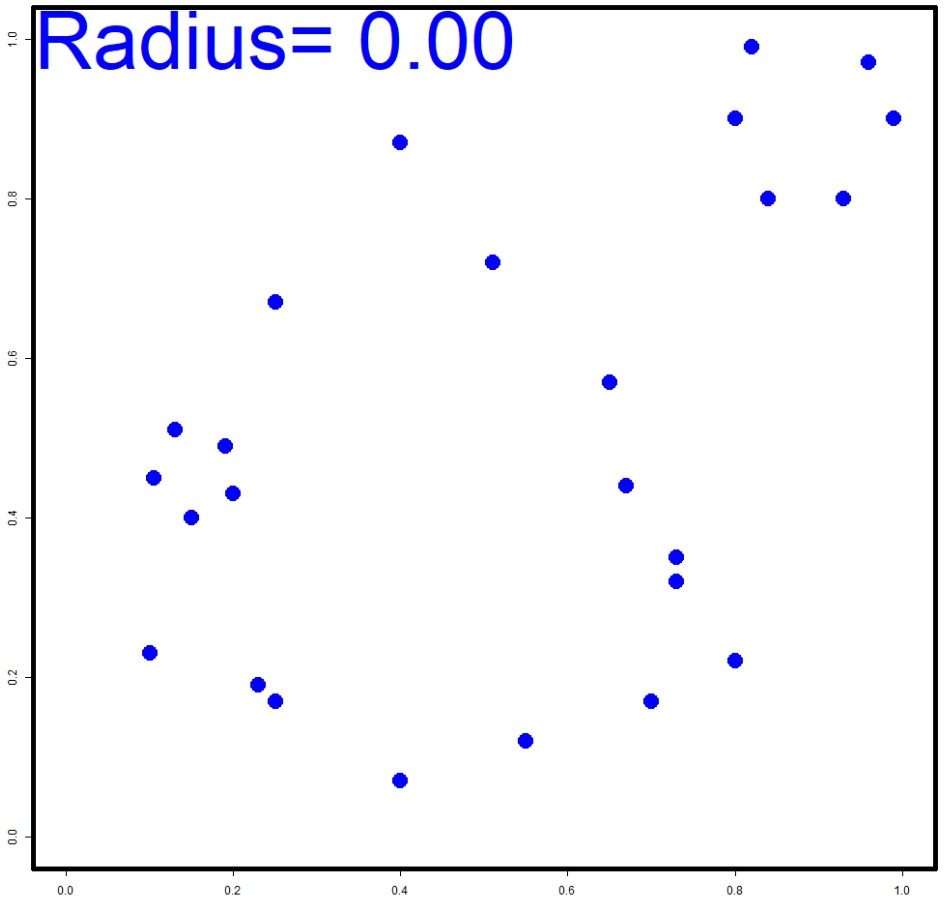


https://en.wikipedia.org/wiki/Ursa_Major

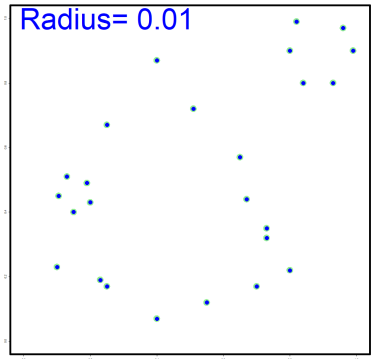
[3] [Edelsbrunner et al., 2000, Zomorodian and Carlsson, 2005]

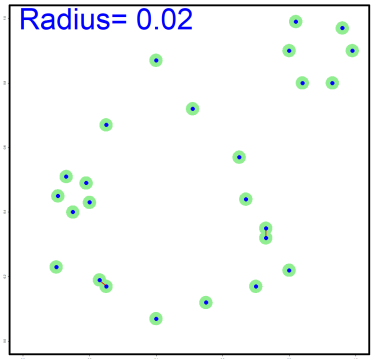
What Is Persistent Homology?

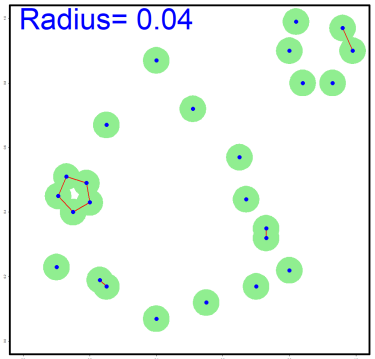
- Decreasing resolution $\Rightarrow$ Data points get closer to each other.
- Any k points that get close enough $\Rightarrow$ Connect them.
- Increasing radius gradually $\Rightarrow$ Components and Holes (Loops) *appear* and *disappear*.

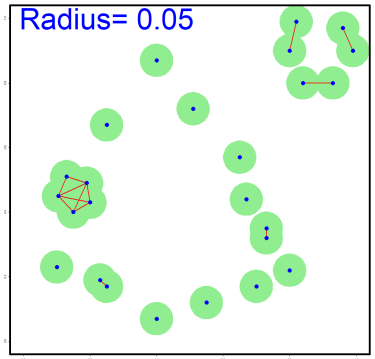


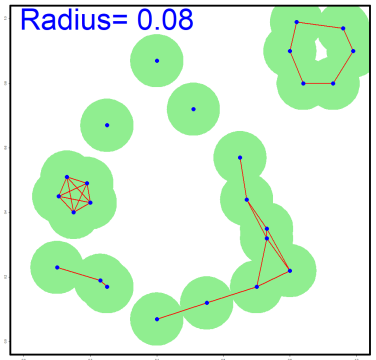
Persistence Diagram Captures Birth and Death Diameters of Holes.

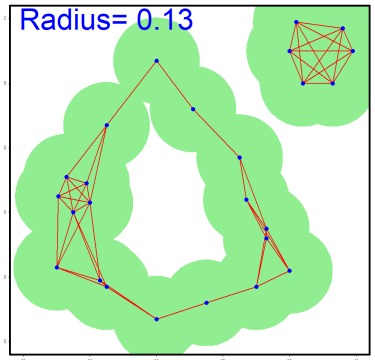


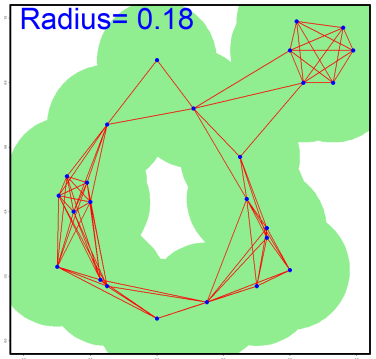


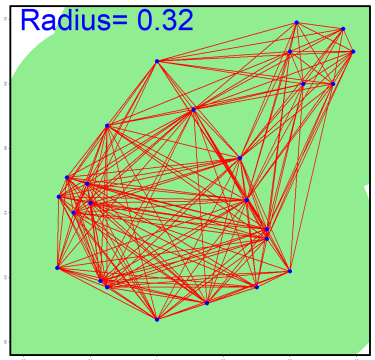


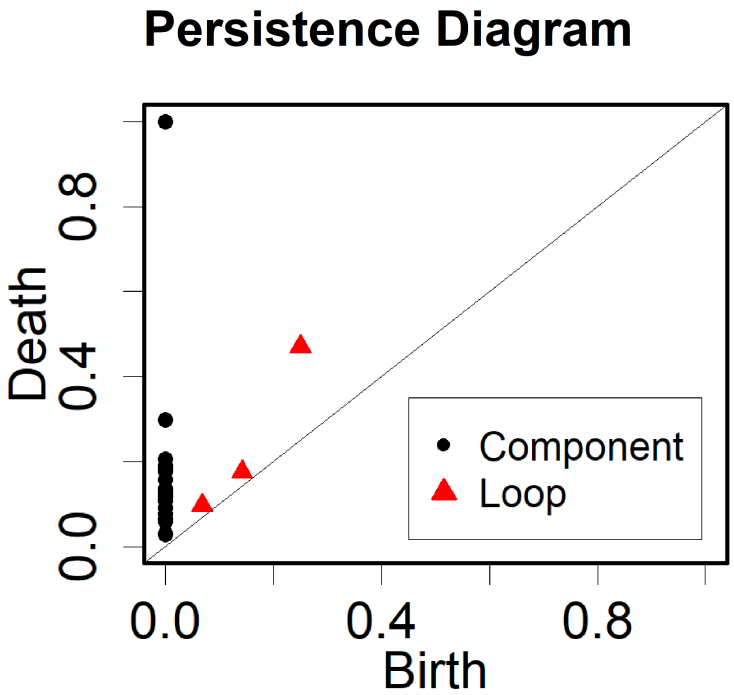




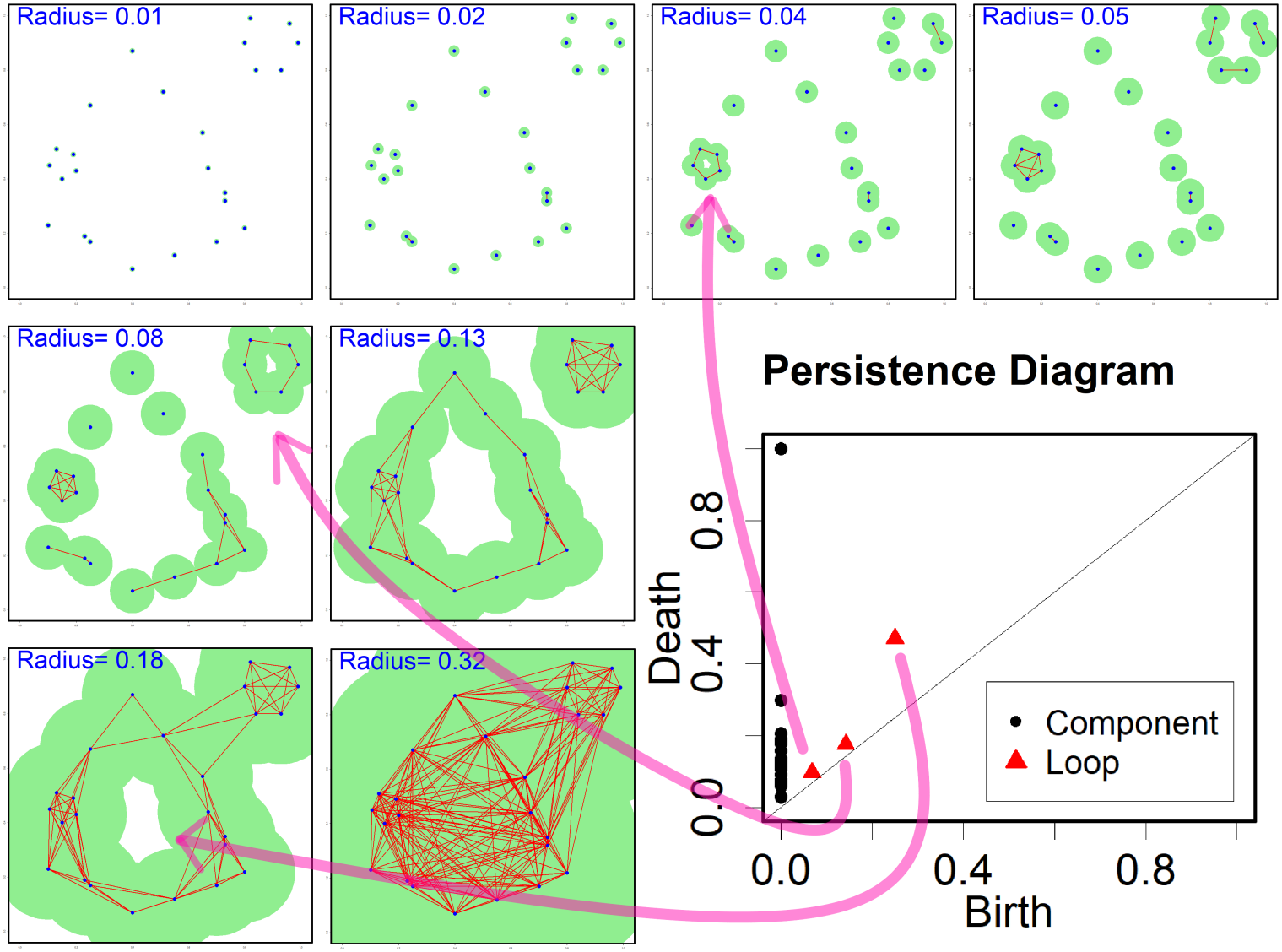






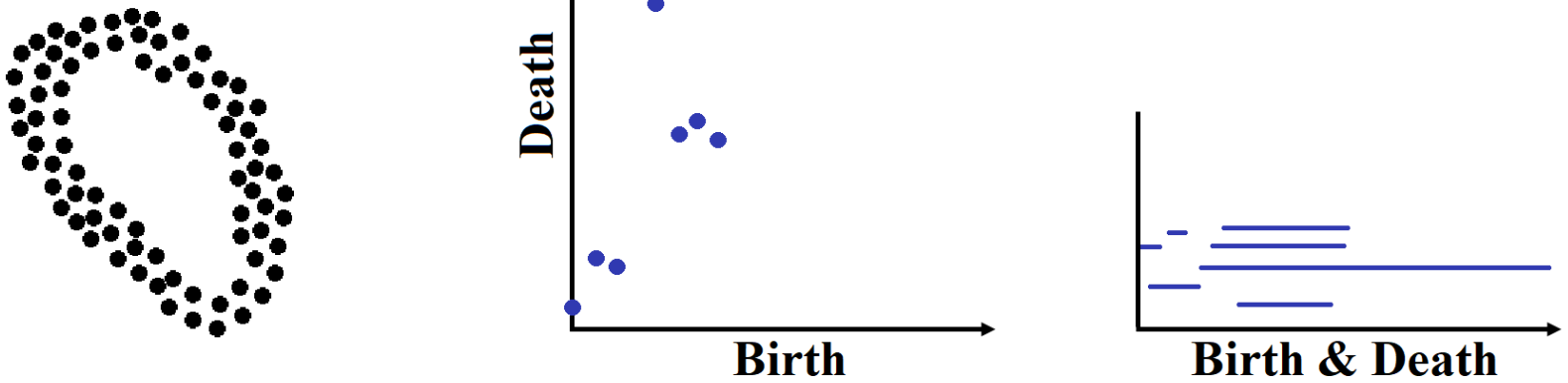


Persistence Diagram Captures Birth and Death Diameters of Holes.



Persistence Diagrams Vs. Barcodes

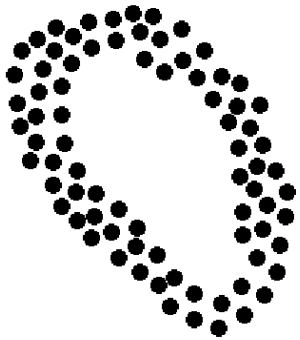
- Persistence diagram: Birth and death of holes shown in 2 dimensions.
- Barcodes [Collins et al., 2004][Ghrist, 2008]: Birth and death of holes shown in 1 dimension.

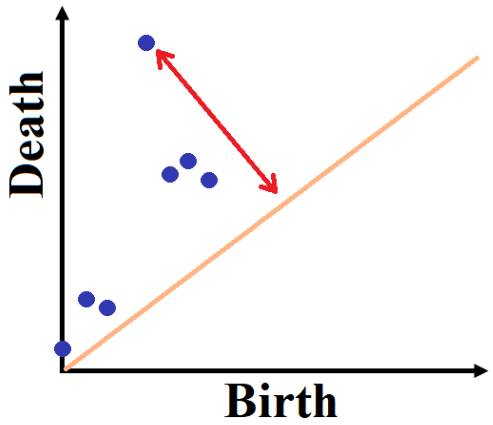


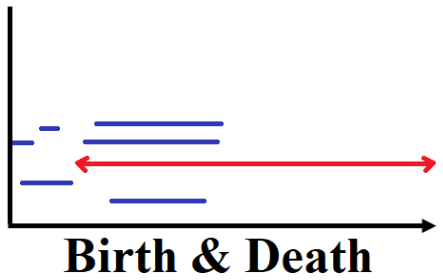
- We will look at the numerical value of barcodes in our work.

Persistence Diagrams Vs. Barcodes

- Persistence diagram: Birth and death of holes shown in 2 dimensions.
- Barcodes [Collins et al., 2004][Ghrist, 2008]: Birth and death of holes shown in 1 dimension.







- We will look at the numerical value of barcodes in our work.

Contents

Positioning TDA in Text Mining

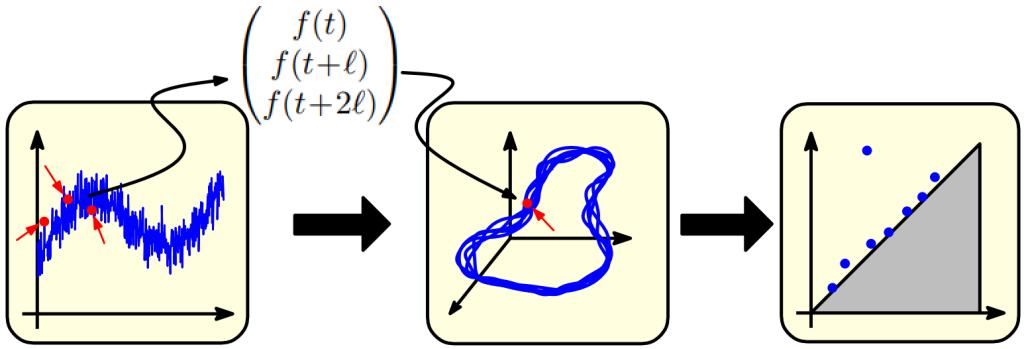
- TDA vs. TF/IDF feature space
- TDA vs. NN
- TDA vs. Word embeddings
- TDA vs. Transformers

TDA in Text Mining: Challenges and Opportunities

- Why Important?
  - Using topological features in addition to the other features
  - Capturing the Order in the text
- Borrowing ideas from Time-Series Analysis
  - Considering Text as n-dimensional time-series of n entities
  - Where vector space representations fail?
- Designing Order-Preserving Text Processing

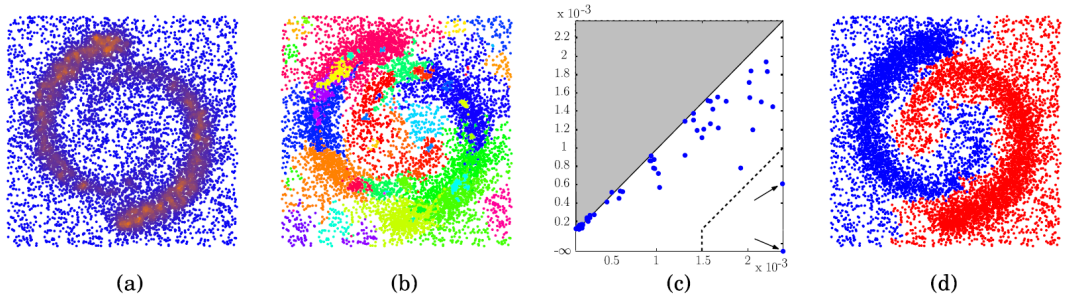
TDA Has Been Applied to Time Series & Signal Processing

- Analysis of Periodic/Quasi-periodic/Recurrent Systems
  - Used Time Delay Embedding. [Skraba et al., 2012]



[Skraba et al., 2012]

- Persistence-Based Clustering on delay embedding [Chazal et al., 2013]



[Chazal et al., 2013]

- Step Detection in Periodic Signals [Khasawneh and Munch, 2018]
- Finding Early Signs of Critical Transitions in Financial Time Series: Stocks/Indices [Gidea, 2017, Gidea and Katz, 2018]

TDA in Text Mining

- Analyzing Discrepancy on Corpus [Wagner et al., 2012]
  - Using vector space representation of corpus.
  - Make Persistence Diagram using Cosine distances.
  - Diagram is a measure of discrepancy on corpus.

- Finding Signs of "Tie-back" in Documents [Zhu, 2013]
  - Divide document to a fixed number of blocks.
  - Apply persistent homology on TF-IDF space of different blocks.

Holes in child writings vs. adolescent writings [Zhu, 2013]

	Child Writing	Adolescent Writing	Adolescent (truncated)
Holes Existence	87%	100%	98%
Total Holes	3.0 ± 0.2	17.6 ± 0.9	3.9 ± 0.2

- [Doshi and Zadrozny, 2018] utilized [Zhu, 2013] algorithm for movie genre detection on the IMDB data set of movie plot summaries.

- [Savle et al., 2019] used TDA on term frequency space for text entailment problem.

Contents

TDA in Text Mining: Challenges and Opportunities

- Why Important?
  - Using topological features in addition to the other features
  - Capturing the Order in the text
- Borrowing ideas from Time-Series Analysis
  - Considering Text as n-dimensional time-series of n entities
- Designing Order-Preserving Text Processing

Methodology

We will use three methods to extract topological features from text.

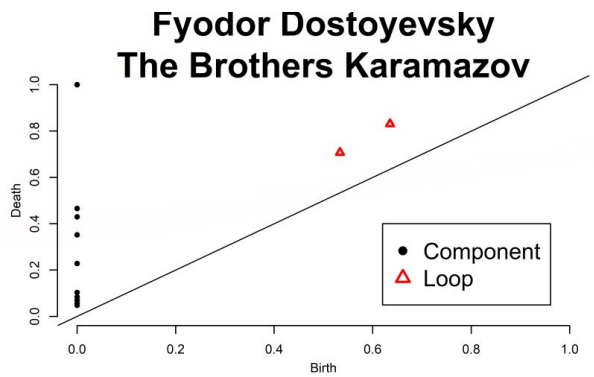
1. Topological features without using conventional representations
2. Topological features from word embeddings space
3. Topological features from TF/IDF space

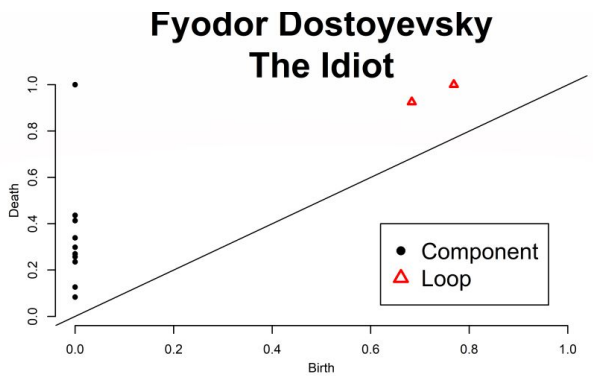
(1) Topological Signature of 19th Century Novelists

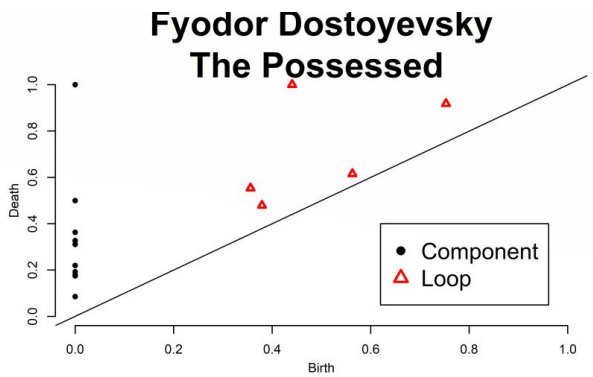
- An application of Persistent Homology in Text Mining
- 75 Novels form Gutenberg.org by 6 novelists
- Predict author solely based on graph of main characters (=persons).
- Average accuracy in binary classification: 77%
- For each Novel:
  - Find positions of each character (person) in the novel.
    - Use Stanford CoreNLP APIs $\rightarrow$ named entity recognizer (NER)
    - Find entities tagged as PERSON.
    - Save place that they appeared (Token Indices).
  - Use only 10 most frequent (important?) characters (persons).
  - Measure the distance between character A and character B.

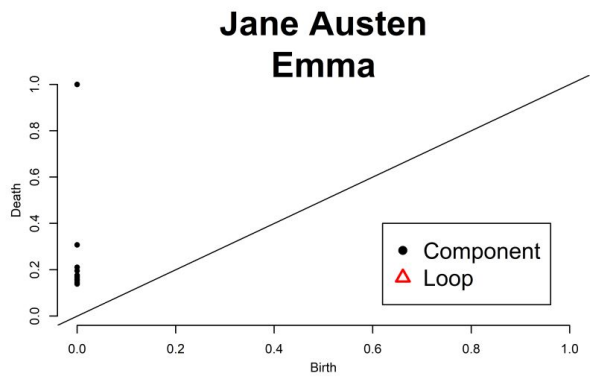
- Using Persistence Diagrams

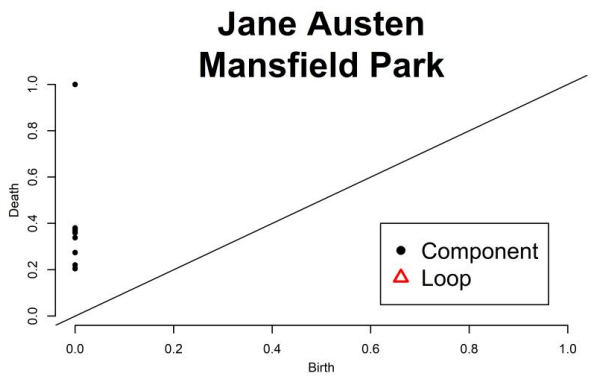
Topological Signature of 19th Century Novelists

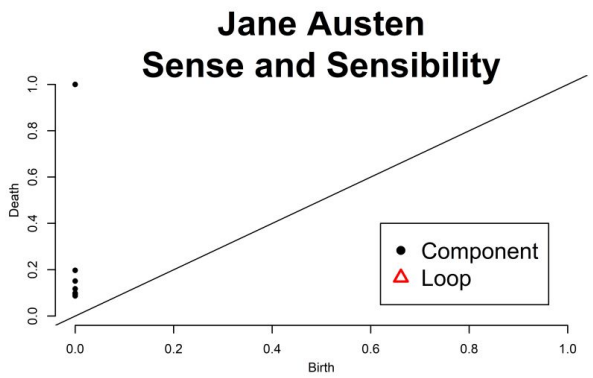


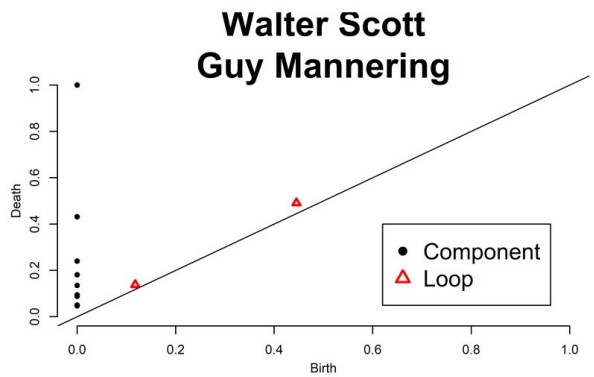


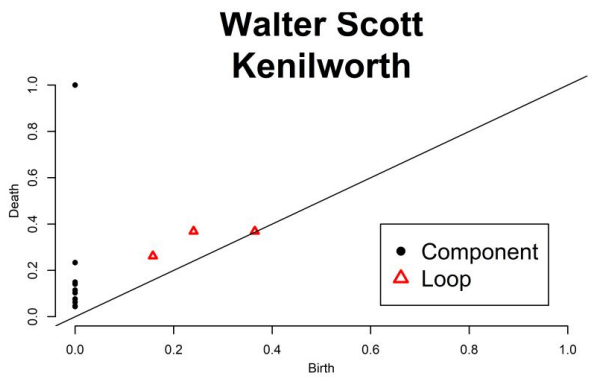


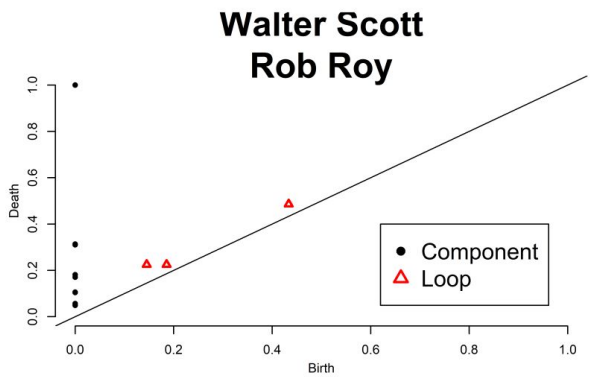




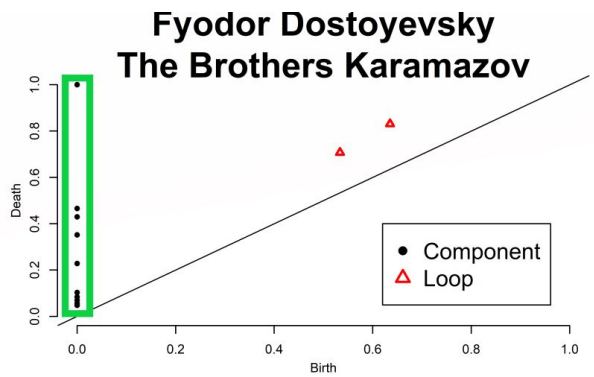


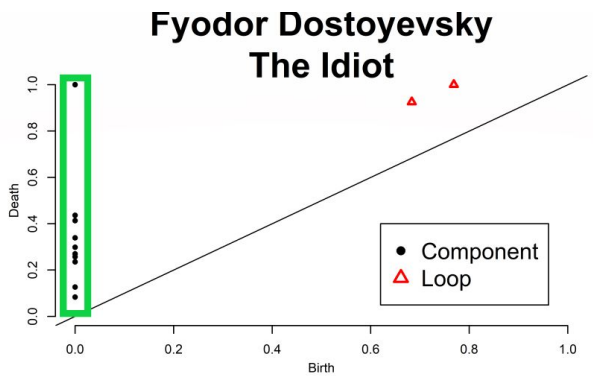


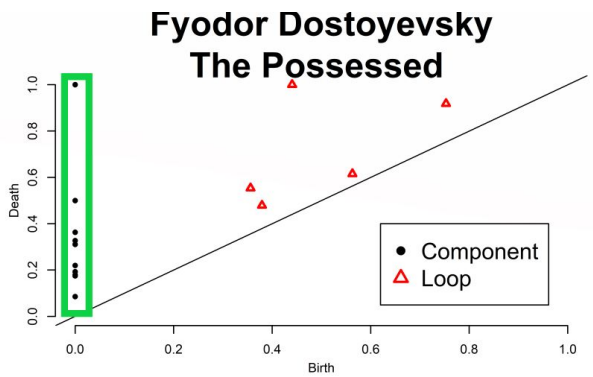


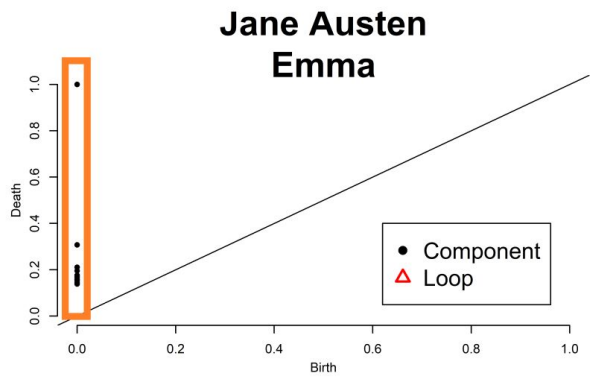


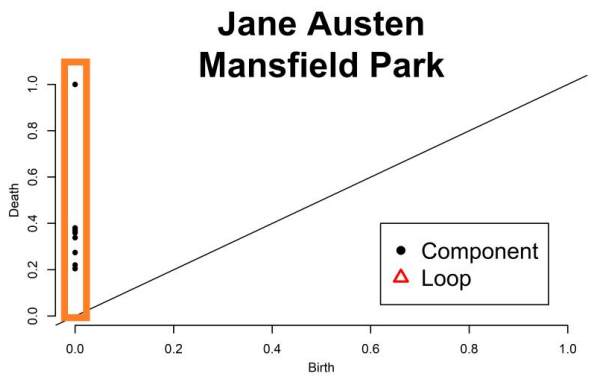
Topological Signature of 19th Century Novelists

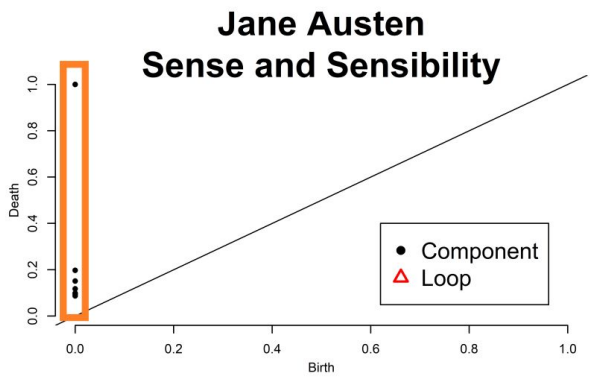


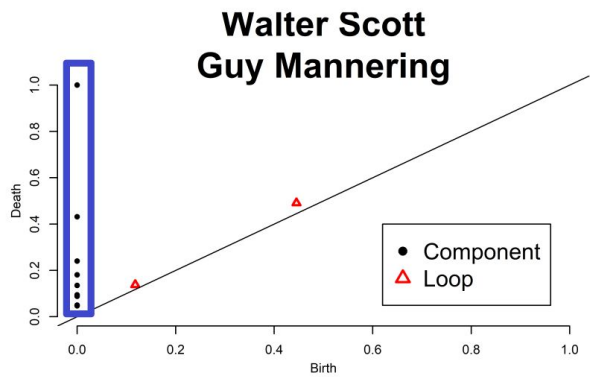


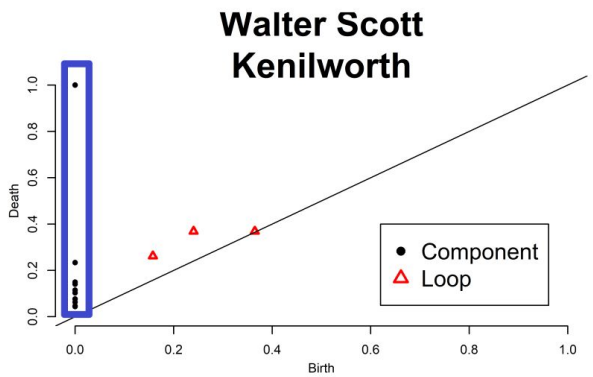


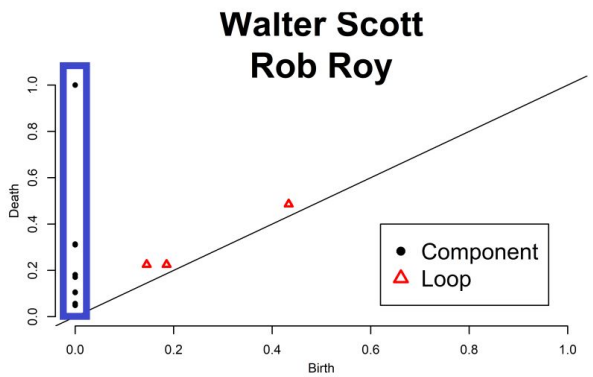




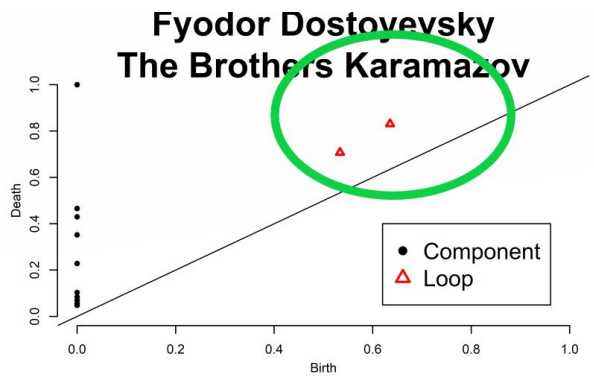


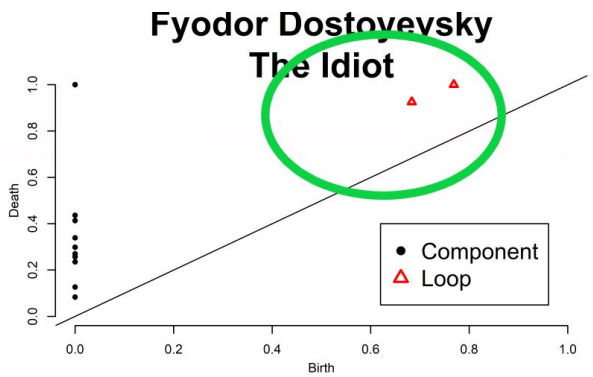


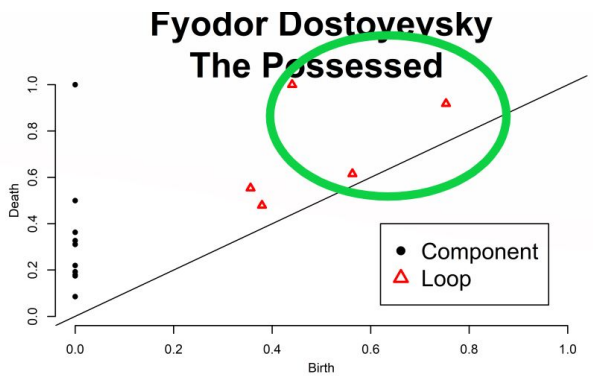


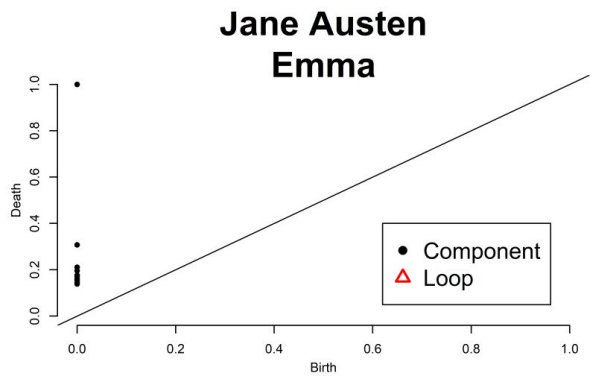


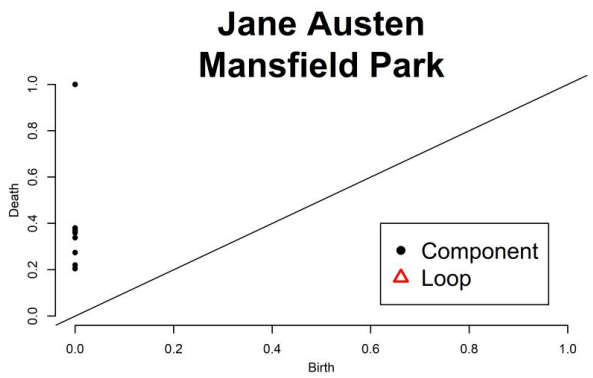
Topological Signature of 19th Century Novelists

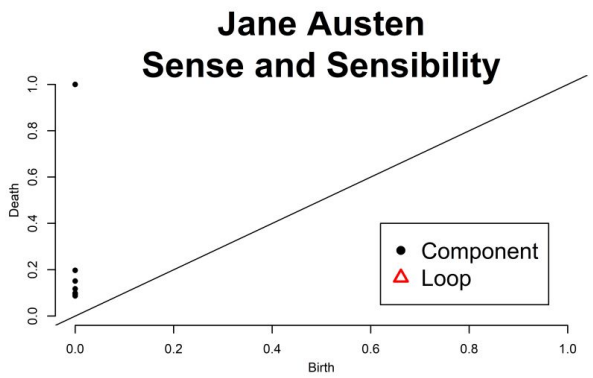


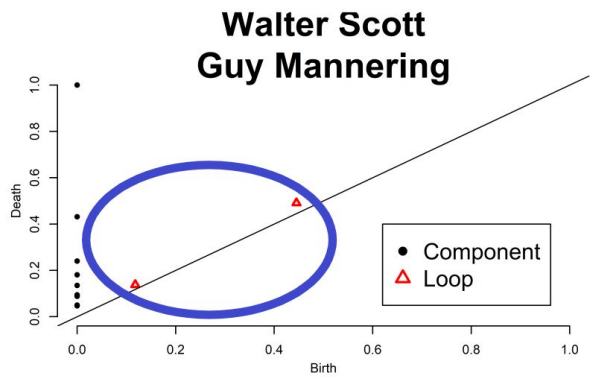


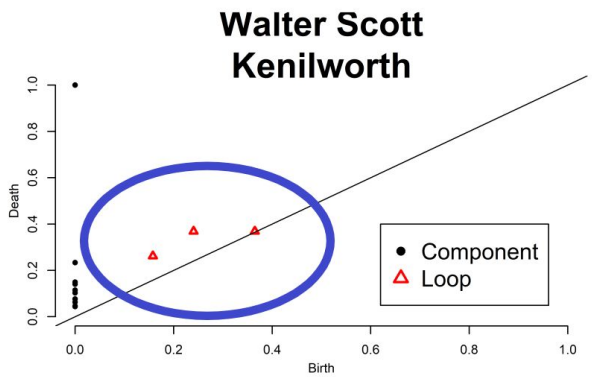


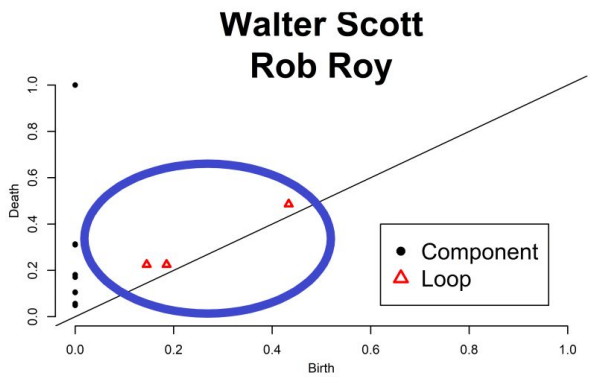












Topological Signature of 19th Century Novelists

- Predicting the author
- Binary Classification (balanced sub-samples)
- 250 times 10-fold cross validation
- 60'000 total predictions
- Using a 5-*NN* algorithm
- Using Wasserstein distance of persistence diagrams

Evaluations (Accuracy)

	Dickens (17)	Zola (18)	Dostoyevsky (8)	Austen (6)	Twain (8)	Scott (18)
C. Dickens	-	87.0	72.2	100.0	74.6	73.9
É. Zola	87.0	-	65.0	64.2	68.8	83.3
Dostoyevsky	72.2	65.0	-	90.2	73.3	55.8
J. Austen	100.0	64.2	90.2	-	82.9	94.7
M. Twain	74.6	68.8	73.3	82.9	-	68.5
W. Scott	73.9	83.3	55.8	94.7	68.5	-
Average	81.5	73.7	71.3	86.4	73.6	75.2

(2) TDA Using Word Embeddings Space

1. TDA without using conventional features
2. TDA using word embeddings
3. TDA using TF-IDF

- In textual documents:
  - Words are discrete. Similar words are not distinguished.
  - "What if we use word embeddings instead of words"?
- Using word embeddings
  - Try pre-trained fastText, Glove, and ConceptNet Numberbatch.
- Using D-dimensional word embedding
  - Text $\rightarrow$ D-dimensional Time Series
- We study the Topology of D-dimensional Time Series of each text.

Word Embedding Representation of a Document

Using word embedding with $D = 300$ dimensions,

$\rightarrow$ a document with T words: $< Word_1, Word_2, \cdots, Word_T >$

$\rightarrow$ can be represented by:

$$\underset{T\times 300}{\text{Genesis}} = \begin{array}{r} \\ \text{in} \\ \text{the} \\ \text{beginning} \\ \text{god} \\ \text{created} \\ \vdots \\ \text{coffin} \\ \text{in} \\ \text{egypt} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{d1} & \text{d2} & \text{d3} & \dots & \text{d300} \end{array} \\ \begin{bmatrix} 0.122 & 0.156 & 0.046 & \dots & -0.034 \\ 0.124 & 0.167 & 0.033 & \dots & -0.026 \\ 0.118 & 0.082 & 0.009 & \dots & 0.010 \\ 0.053 & 0.040 & -0.016 & \dots & 0.134 \\ 0.110 & 0.035 & -0.003 & \dots & -0.029 \\ \vdots & \vdots & \vdots & & \vdots \\ 0.035 & 0.019 & 0.110 & \dots & 0.025 \\ 0.122 & 0.156 & 0.046 & \dots & -0.034 \\ -0.094 & 0.043 & 0.014 & \dots & -0.013 \end{bmatrix} \end{array}$$

A Novel Algorithm for Document Representation

- A textual Doc: A D-dimensional time series (intput)
- Define some distance among D embedding dimensions.
  - Based on Cosine Similarity, Correlation, or Covariance
- For each document:
  - $\Rightarrow$ Make a graph of D vertices.
  - $\Rightarrow$ Get the persistent diagram of the graph.
  - $\Rightarrow$ How much the PD will change if we exclude dimension d?
    - $\Rightarrow$ Do it for $d = 1, \ldots, D$.
    - $\Rightarrow$ "Differentiate" the PD's using Wasserstein distance.
  - $\Rightarrow$ Get D topological features (output) for each document.

What Is the Algorithm Doing?

- For each document:

  $\Rightarrow$ Build persistent **diagram**.

  $\Rightarrow$ Get the **sensitivity** of PD to each embedding dimension.

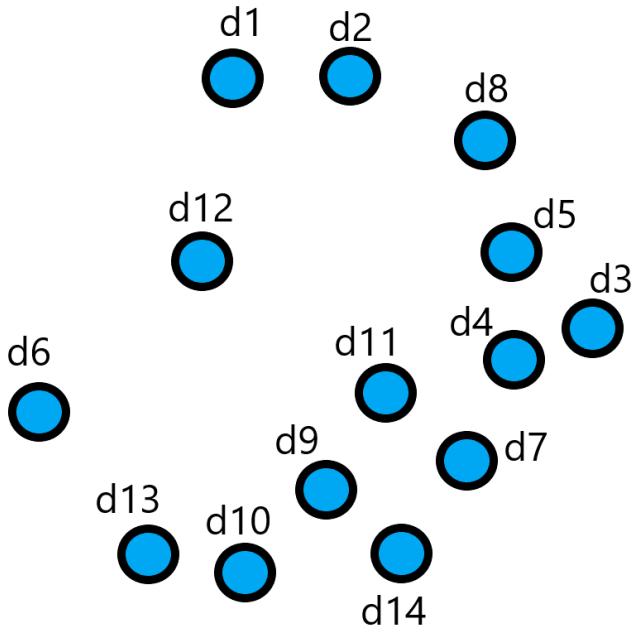
  $\Rightarrow$ Use it as the **sensitivity** of the **document itself** to that embedding dimension.

- Use these features for classification, clustering, etc.

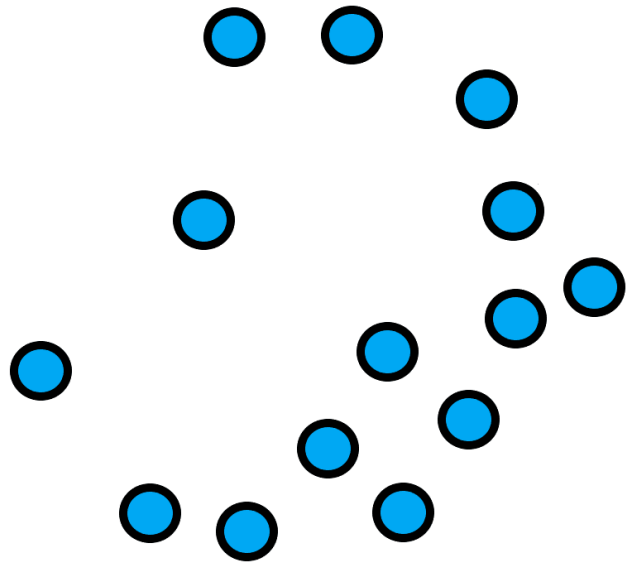
- Is the final result a "Document Embedding"?

Why Differentiating Persistent Diagram?

Coordinates (Graph) of embedding dimensions



Graph as PH perceives



Persistent homology does not distinguish the order of dimensions if we do not differentiate the results.

Toward Order Preserving Algorithm

- We define the distance between embedding dimensions based on Covariance/Correlation/Cosine Matrix $\longrightarrow$ It is not order preserving.
  - Like a bag-of-words model, shuffling the words produces same results.

- What if we smooth the time-series first? After smoothing, each index of an embedding dimension is being compared to:

  (1) the same index of other dimensions
  (2) a few lags/leads of other dimensions

$$
\begin{bmatrix}
X_1(1) & \cdots & X_d(1) & \cdots & X_D(1) \\
\vdots & & \vdots & & \vdots \\
X_1(t-2) & \cdots & X_d(t-2) & \cdots & X_D(t-2) \\
X_1(t-1) & \cdots & X_d(t-1) & \cdots & X_D(t-1) \\
X_1(t) & \cdots & X_d(t) & \cdots & X_D(t) \\
X_1(t+1) & \cdots & X_d(t+1) & \cdots & X_D(t+1) \\
X_1(t+2) & \cdots & X_d(t+2) & \cdots & X_D(t+2) \\
\vdots & & \vdots & & \vdots \\
X_1(T) & \cdots & X_d(T) & \cdots & X_D(T)
\end{bmatrix}_{T \times D}
$$

Experiment to Examine TDA on Word Embeddings

- **arXiv Papers**: We downloaded all of arXiv papers in quantitative finance[4] published between 2011 and 2018.
  - We selected five major categories (subject tags): General Finance, Statistical Finance, Mathematical Finance, Pricing of Securities, and Risk Management.
  - We topological features to XGBoost to classify arXiv topics.
  - Achieved F1-score of 0.643.

    $\longrightarrow$ Outperforming our best CNN classifier where F1-score was 0.607.

- **IMDB Movie Review** [Maas et al., 2011]: Using IMDB reviews annotated by positive/negative labels, we examined the topological algorithm on word embeddings for binary sentiment classification.
  - Fed topological features to XGBoost to classify movie review polarity:
  - Achieved F1-score of 0.884.

    $\longrightarrow$ Slightly better than the best previous results where F1 was 0.880.

[4] https://arXiv.org/archive/q-fin

(3) TDA Using TF-IDF Space

1. TDA without using conventional features
2. TDA using word embeddings
3. TDA using TF-IDF

Similar to the [Zhu, 2013] method, the idea is to:

(a) Dividing the textual document to a fixed number of blocks.

(b) Analyze the cosine distances among different blocks.

(c) Search for repetitive patters among the blocks.

The results might be not as strong as TDA features from WE. But we may use them in addition to other features for text classification.

TDA Using TF-IDF: Details of the Algorithm

For each document:

1. Divide it to 10 blocks.
2. Calculate the TF-IDF of each block.
3. Calculate the Cosine similarity among different blocks.

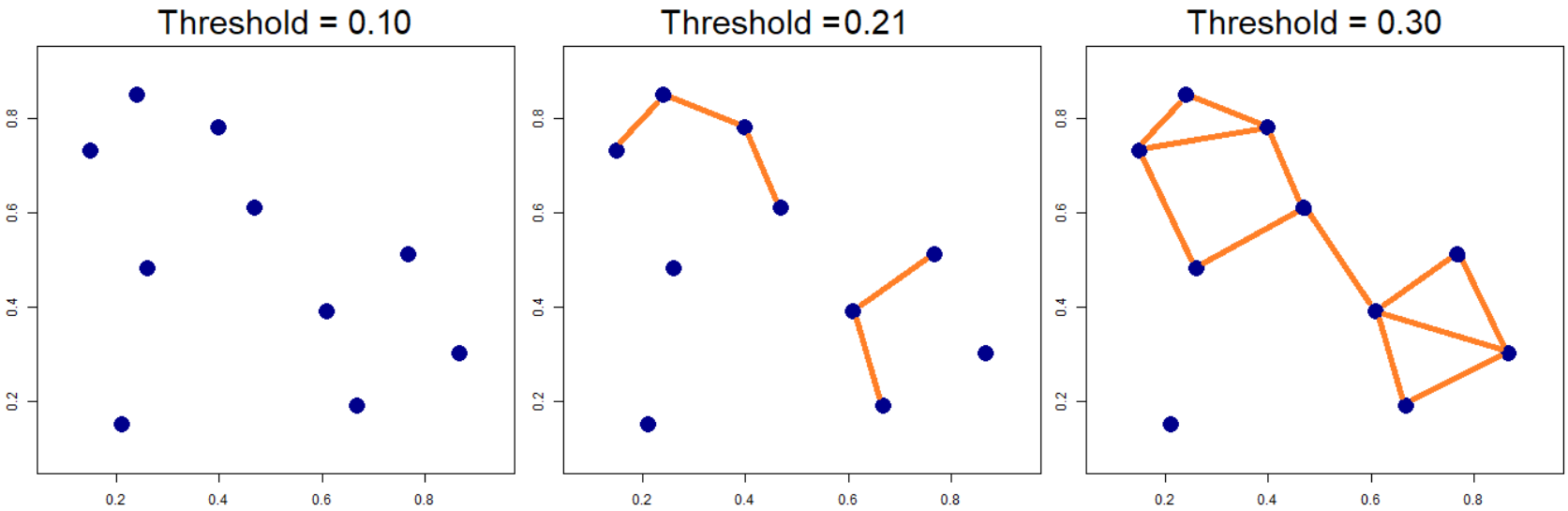
   $\Rightarrow$ We have a weighted graph whose vertices are 10 text blocks.

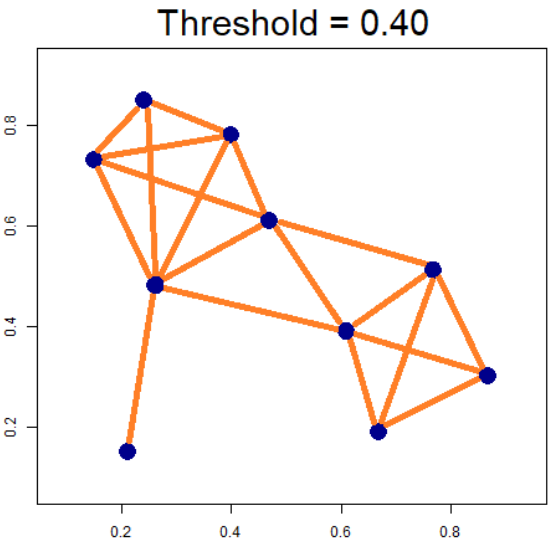
4. Apply persistent homology on the graph.

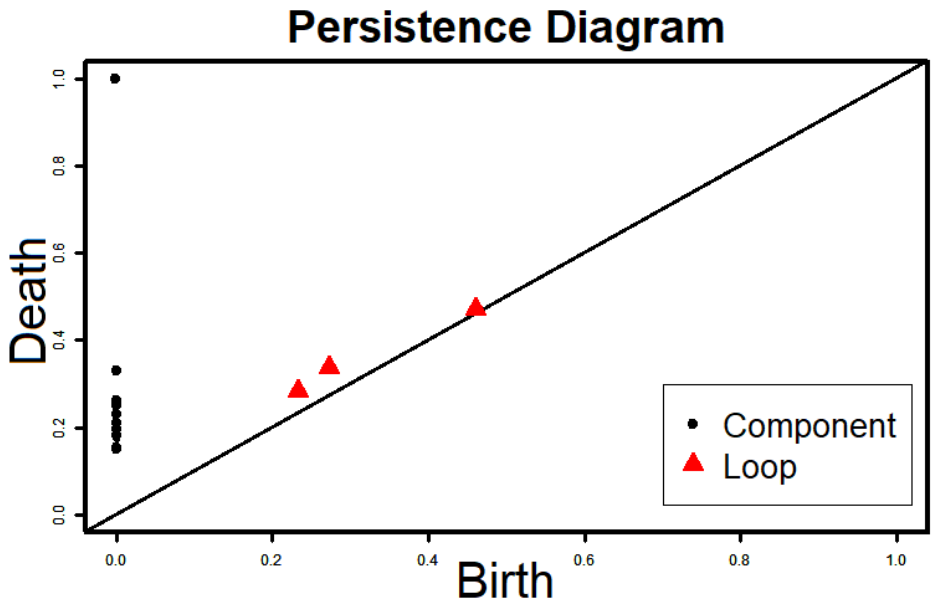
   $\Rightarrow$ We will get birth/death diameters for dimension 0 (components).

   $\Rightarrow$ We will get birth/death diameters for dimension 1 (loops)

TDA Using TF-IDF: Details of the Algorithm







TDA Using TF-IDF: Details of the Algorithm

For each document:

- In dimension 0 (components):
  - All the birth diameters are always zero.
  - We always get 9 diameter of death. $\Rightarrow$ We get 9 features.
- In dimension 1 (loops):
  - We may see different number of loops for different documents.
  - How to flatten the results? A trick: Retrieve only 5 statistics.
    1. number of loops
    2. average diameter of birth
    3. average diameter of duration (duration = death - birth)
    4. standard deviation of of birth diameters
    5. standard deviation of duration diameters
- Totally, we get $9 + 5 = 14$ features for the document.

Evaluation of TDA on TF-IDF & Word Embeddings

- We run both algorithms for (1) word embeddings and (2) TF-IDF on Wikipedia Movie Plots from Kaggle. Selected four major genres.

- Fed the topological features to XGBoost to predict the genres. Also tried BiLSTM to benchmark the results.

	Classifier	Pre.	Rec.	F1	Acc.
1	BiLSTM	68.0	59.7	0.608	76.2
2	XGBoost on TP1	59.6	53.2	0.560	71.1
3	XGBoost on TP1 & TP2	59.9	53.7	0.564	71.4
4	BiLSTM + XGBoost on TP1	67.8	64.8	0.656	77.3
5	BiLSTM + XGBoost on TP1 & TP2	68.5	64.6	**0.659**	**77.8**

- TP1: Topological features from Word embeddings.
- TP2: Topological features from TF-IDF.

- *Topological features from word embeddings are much more helpful than topological features from TF-IDF.*

Contents

Limitations ≈ Opportunities

- Only classification here, and some *inference* ([Savle et al., 2019])
- What about *summarization*?
  – Embeddings represent semantic/conceptual connections, and topology compresses. Could this be helpful for summarization?
- What about other data sets?
- Newest *attention models* not investigated
  – In principle NN can find topological features of any data
- `Mapping back from topology to text:`
  – Poorly understood reasons for improvements
  – Simplexes as a representation of concept boundaries?
  – Principle-based compression?

Summary

- We introduced and evaluated three methods of extracting topological representations from text:
  1. using TF-IDF vector space,
  2. using word embeddings space, and
  3. using name entities without any conventional features.
- Topological representation extracted without using conventional features is primarily useful for author profiling/classification.
- Features extracted from word embeddings showed the best performance, while features from TF-IDF are primarily designated to find repetitive patterns in text.
- TDA features carrying exclusive information that is not reflected in conventional features. They can boost the classification results.

Future Direction

- Analyze the co-appearances of more entity types. We analyzed only the name entities tagged as 'person'. We can extend it to locations, POS tags, etc.
- We mainly focused on text classification. We will investigate of extend and apply our methods for other natural language processing tasks, such as summarization or question answering.
- Apply persistent homology on attention models (the attention matrices).
- A open problem, explainability: Find the actual text behind the topological structures.

Questions

Questions?

Bibliography I

Adams, H. and Carlsson, G. (2015).

Evasion paths in mobile sensor networks.

The International Journal of Robotics Research, 34(1):90–104.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016).

Enriching word vectors with subword information.

arXiv preprint arXiv:1607.04606.

Bubenik, P. (2015).

Statistical topological data analysis using persistence landscapes.

The Journal of Machine Learning Research, 16(1):77–102.

Chazal, F., Guibas, L. J., Oudot, S. Y., and Skraba, P. (2013).

Persistence-based clustering in riemannian manifolds.

Journal of the ACM (JACM), 60(6):41.

Collins, A., Zomorodian, A., Carlsson, G., and Guibas, L. J. (2004).

A barcode shape descriptor for curve point cloud data.

Computers & Graphics, 28(6):881–894.

Das, S. and DebBarma, M. K. (2018).

Hole detection in wireless sensor network: A review.

In *Recent Findings in Intelligent Computing Techniques*, pages 87–96. Springer.

De Silva, V. and Ghrist, R. (2006).

Coordinate-free coverage in sensor networks with controlled boundaries via homology.

The International Journal of Robotics Research, 25(12):1205–1222.

Bibliography II

Doshi, P. and Zadrozny, W. (2018).

Movie genre detection using topological data analysis.

In *International Conference on Statistical Language and Speech Processing*, pages 117–128. Springer.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000).

Topological persistence and simplification.

In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE.

Ghrist, R. (2008).

Barcodes: the persistent topology of data.

Bulletin of the American Mathematical Society, 45(1):61–75.

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., and Chatzisavvas, K. C. (2017).

Sentiment analysis leveraging emotions and word embeddings.

Expert Systems with Applications, 69:214–224.

Gidea, M. (2017).

Topological data analysis of critical transitions in financial networks.

In *International Conference and School on Network Science*, pages 47–59. Springer.

Gidea, M. and Katz, Y. (2018).

Topological data analysis of financial time series: Landscapes of crashes.

Physica A: Statistical Mechanics and its Applications, 491:820–834.

Guss, W. H. and Salakhutdinov, R. (2018).

On characterizing the capacity of neural networks using algebraic topology.

arXiv preprint arXiv:1802.04443, pages 1–13.

Bibliography III

Hofer, C., Kwitt, R., Niethammer, M., and Dixit, M. (2019).
Connectivity-optimized representation learning via persistent homology.
In *International Conference on Machine Learning*, pages 2751–2760.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016).
Bag of tricks for efficient text classification.
arXiv preprint arXiv:1607.01759.

Khasawneh, F. A. and Munch, E. (2018).
Topological data analysis for true step detection in periodic piecewise constant signals.
Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474(2218):20180027.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011).
Learning word vectors for sentiment analysis.
In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Mittal, K. and Gupta, S. (2017).
Topological characterization and early detection of bifurcations and chaos in complex systems using persistent homology.
Chaos: An Interdisciplinary Journal of Nonlinear Science, 27(5):051102.

Naitzat, G., Zhitnikov, A., and Lim, L.-H. (2020).
Topology of deep neural networks.
arXiv preprint arXiv:2004.06093.

Pennington, J., Socher, R., and Manning, C. (2014).
Glove: Global vectors for word representation.
In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Bibliography IV


Petri, G., Scolamiero, M., Donato, I., and Vaccarino, F. (2013).
Topological strata of weighted complex networks.
PloS one, 8(6):e66506.


Savle, K., Zadrozny, W., and Lee, M. (2019).
Topological data analysis for discourse semantics?
In *Proceedings of the 13th International Conference on Computational Semantics-Student Papers*, pages 34–43.


Shaukat, Z., Zulfiqar, A. A., Xiao, C., Azeem, M., and Mahmood, T. (2020).
Sentiment analysis on imdb using lexicon and neural networks.
SN Applied Sciences, 2(2):1–10.


Skraba, P., de Silva, V., and Vejdemo-Johansson, M. (2012).
Topological analysis of recurrent systems.
In *NIPS 2012 Workshop on Algebraic Topology and Machine Learning, December 8th, Lake Tahoe, Nevada*, pages 1–5.


Speer, R., Chin, J., and Havasi, C. (2017).
Conceptnet 5.5: An open multilingual graph of general knowledge.
In *Thirty-First AAAI Conference on Artificial Intelligence*.


Wagner, H., Dłotko, P., and Mrozek, M. (2012).
Computational topology in text mining.
In *Computational Topology in Image Context*, pages 68–78. Springer.


Zhu, X. (2013).
Persistent homology: An introduction and a new text representation for natural language processing.
In *IJCAI*, pages 1953–1959.

Bibliography V

Zomorodian, A. and Carlsson, G. (2005).
Computing persistent homology.
Discrete & Computational Geometry, 33(2):249–274.

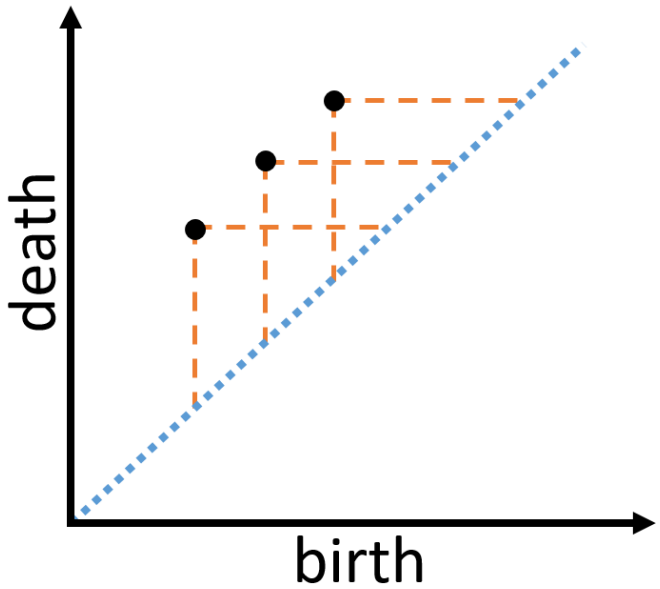
Persistent Landscape

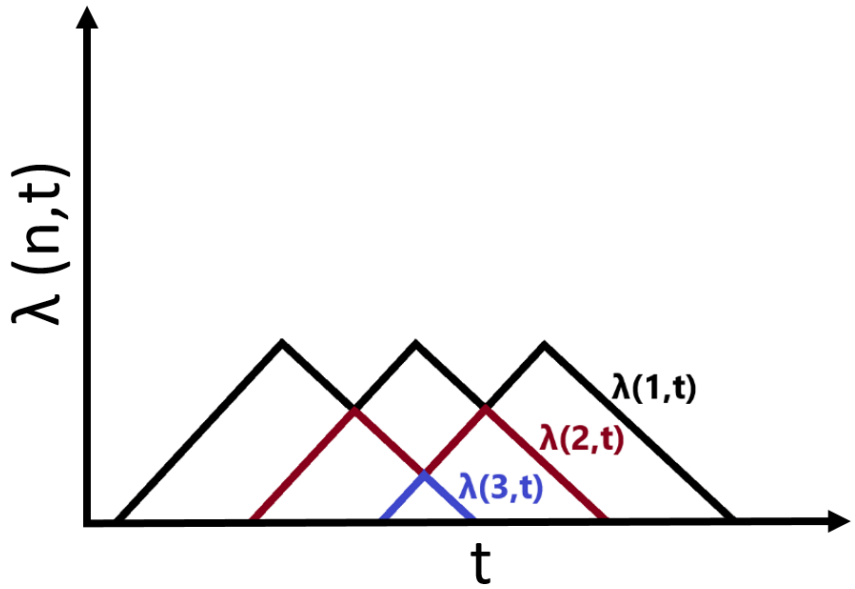
- Persistent Landscape [Bubenik, 2015]: Real-valued function
- Intuitively rotate persistence diagram by $\pi/4$

$\lambda\colon \mathbb{N} \times \mathbb{R} \to \mathbb{R}$

$\lambda(n, t) = sup\{radius \geq 0 \mid \beta(t - radius\ ,\ t + radius) \geq n\}$

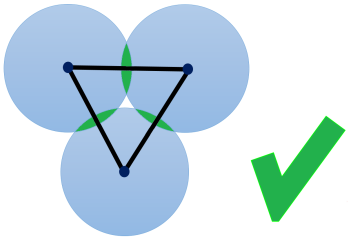
$\forall n \in \mathbb{N}$



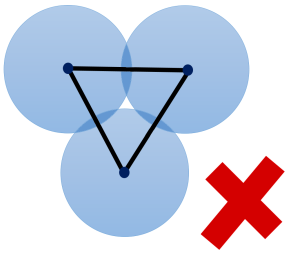


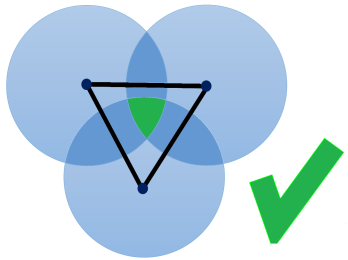
Information Structure & Filtration

- *Rips Filtration* [Ghrist, 2008]: A k-simplex has k nodes with pairwise distance $\leq \epsilon$.



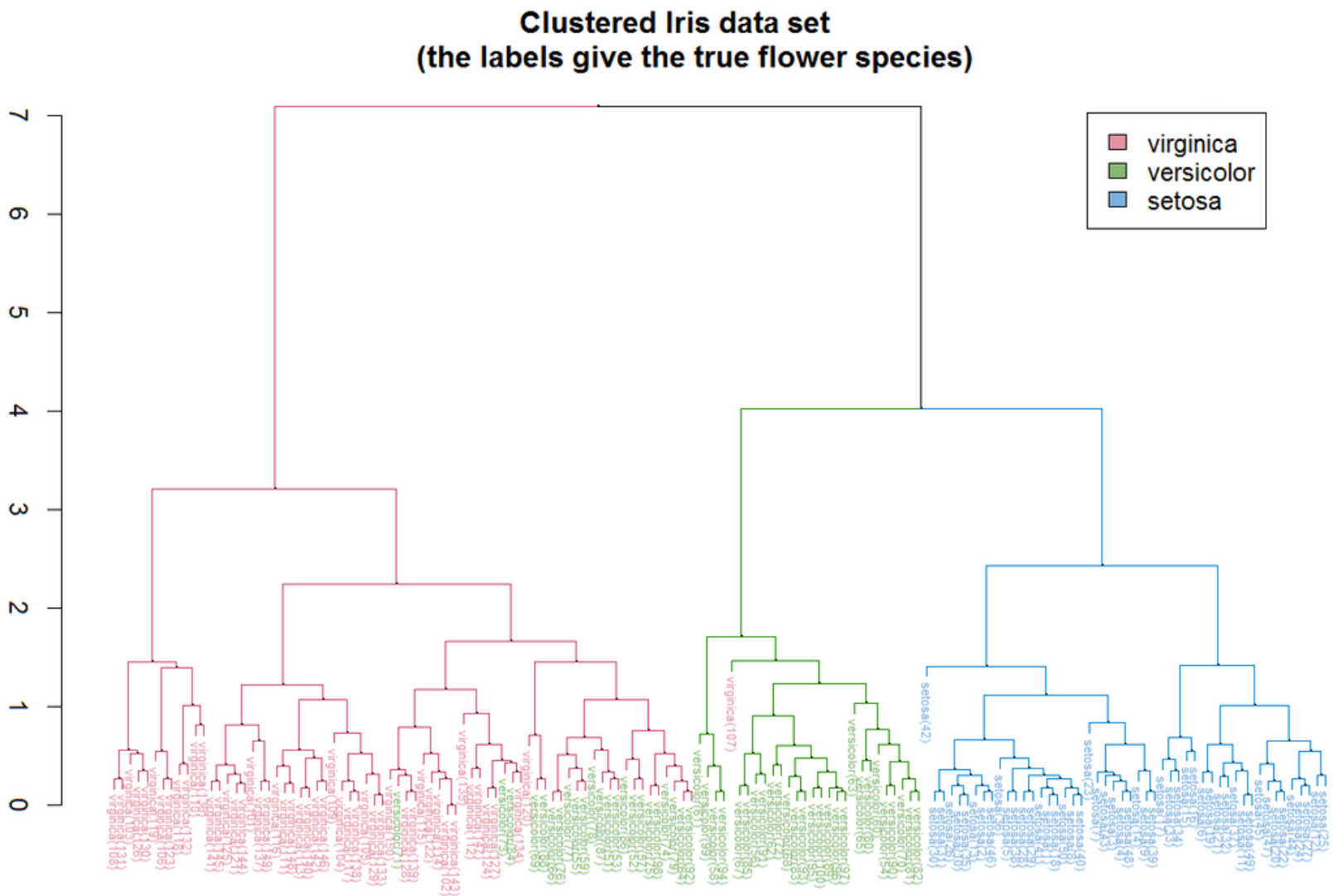
- *Čech Complex*: Slightly stronger conditions: The regions around the nodes of a simplex within the radii equal to the ϵ altogether should have a non-empty intersection.





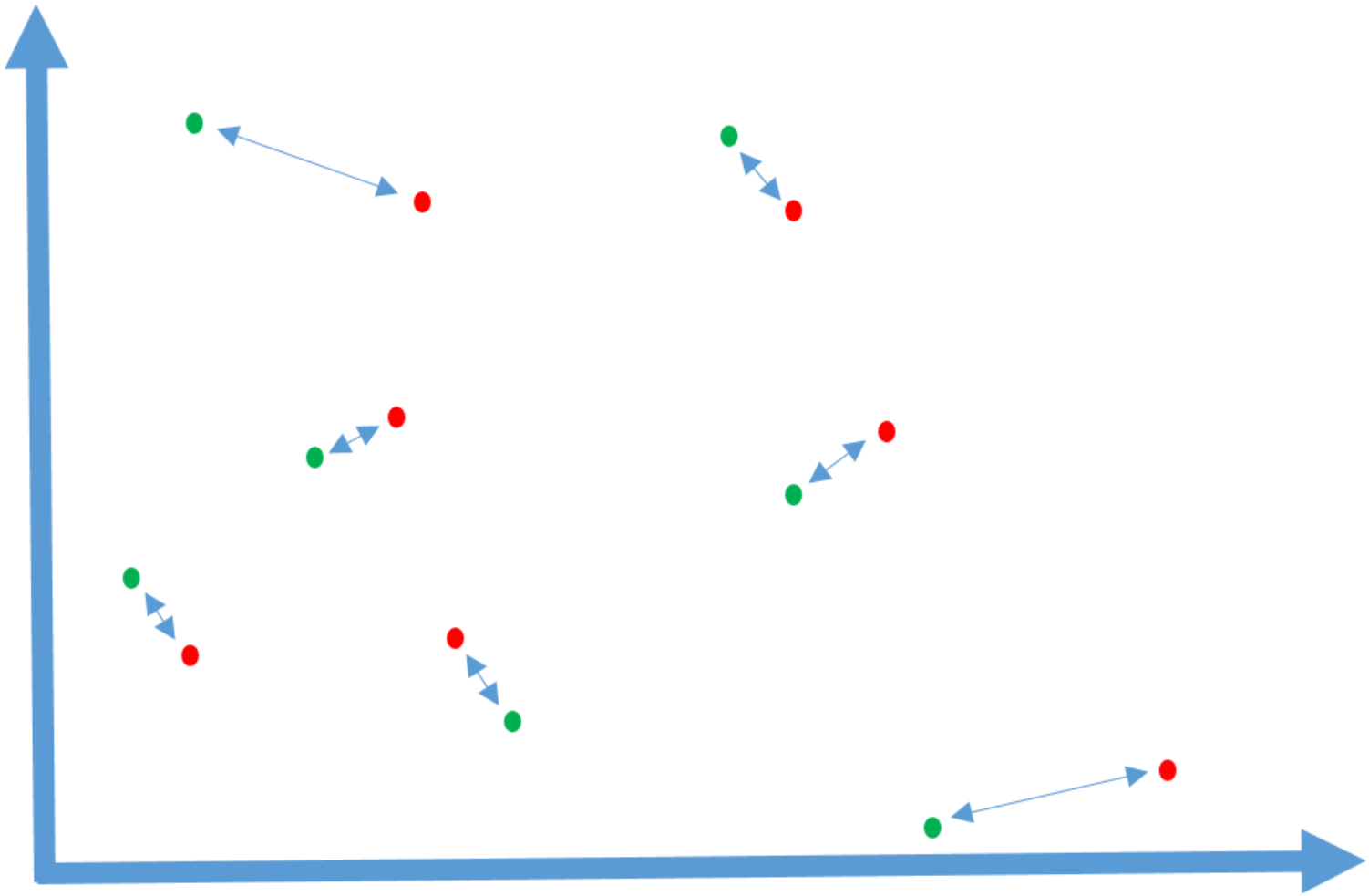
- *Weight Rank Clique Filtration* [Petri et al., 2013]: Dealing with a weighted graph (instead of a data cloud): Threshold the weights and increase the weights threshold gradually.

Dendrogram



https://en.wikipedia.org/wiki/Dendrogram

Wasserstein Distance: Intuition



How to define distances?

Distance of character A and character B

$$Distance_t(\mathrm{A}, \mathrm{B}) = WD_{0.5}(\tilde{I}^{(1+t)}, \tilde{J}^{(1+t)})$$

- $\tilde{I}, \tilde{J}$: normalized indices of positions where A and B appear respectively
- $t = 0 \rightarrow$ Wasserstein distance of order 0.5 of $\tilde{I}, \tilde{J}$
- Order 0.5 $\rightarrow$ sensitive to the closer element-wise distances
- Where they co-appear? $t = 0, -\epsilon, and + \epsilon$

Distance of novel X and novel Y

$$Distance_t(\mathrm{X}, \mathrm{Y}) = WD\{PD_t^0(X), PD_t^0(Y)\} \; + \; WD\{PD_t^1(X), PD_t^1(Y)\}$$

$$Distance(\mathrm{X}, \mathrm{Y}) = \{ \sum_{t \in \{-\epsilon, 0, +\epsilon\}} Distance_t(X, Y)^2 \}^{\frac{1}{2}}$$

Toward Order Preserving Algorithm

- Covariance/Correlation/Cosine Matrix is not order preserving.
  - Like a bag-of-words model, shuffling the words produces same results.
- What if we use a smoothed time-series?
  - Using exponential smoothing?
  - Using local averages (sliding window of size $\omega = 5$):

$$\tilde{X}_i = \tilde{X}_i(t) = X_i(t-2) + X_i(t-1) + \cdots + X_i(t+2)$$

Toward Order Preserving Algorithm

- Covariance/Correlation/Cosine Matrix is not order preserving.
  - Like a bag-of-words model, shuffling the words produces same results.
- What if we use a smoothed time-series?
  - Using exponential smoothing?
  - Using local averages (sliding window of size $\omega = 5$):

$$\tilde{X}_i = \tilde{X}_i(t) = X_i(t-2) + X_i(t-1) + \cdots + X_i(t+2)$$

$$\begin{aligned}
\mathbb{E}[\tilde{X}_i \tilde{X}_j] &= \mathbb{E}[\tilde{X}_i(t)\tilde{X}_j(t)] \\
&\approx 5\mathbb{E}[X_i(t)X_j(t)] \\
&+ 4\mathbb{E}[X_i(t-1)X_j(t)] + 4\mathbb{E}[X_i(t)X_j(t-1)] \\
&+ 3\mathbb{E}[X_i(t-2)X_j(t)] + 3\mathbb{E}[X_i(t)X_j(t-2)] \\
&+ 2\mathbb{E}[X_i(t-3)X_j(t)] + 2\mathbb{E}[X_i(t)X_j(t-3)] \\
&+ 1\mathbb{E}[X_i(t-4)X_j(t)] + 1\mathbb{E}[X_i(t)X_j(t-4)]
\end{aligned}$$

Toward Order Preserving Algorithm

- Covariance/Correlation/Cosine Matrix is not order preserving.
  - Like a bag-of-words model, shuffling the words produces same results.
- What if we use a smoothed time-series?
  - Using exponential smoothing?
  - Using local averages (sliding window of size $\omega = 5$):

$$\tilde{X}_i = \tilde{X}_i(t) = X_i(t-2) + X_i(t-1) + \cdots + X_i(t+2)$$

$$\begin{aligned}
\mathbb{E}[\tilde{X}_i \tilde{X}_j] &= \mathbb{E}[\tilde{X}_i(t)\tilde{X}_j(t)] \\
&\approx 5\mathbb{E}[X_i(t)X_j(t)] \\
&+ 4\mathbb{E}[X_i(t-1)X_j(t)] + 4\mathbb{E}[X_i(t)X_j(t-1)] \\
&+ 3\mathbb{E}[X_i(t-2)X_j(t)] + 3\mathbb{E}[X_i(t)X_j(t-2)] \\
&+ 2\mathbb{E}[X_i(t-3)X_j(t)] + 2\mathbb{E}[X_i(t)X_j(t-3)] \\
&+ 1\mathbb{E}[X_i(t-4)X_j(t)] + 1\mathbb{E}[X_i(t)X_j(t-4)]
\end{aligned}$$

Considering n-grams with weights

Distance among Embedding Dimensions

- How to Define Distance between Embedding Dimensions?

$$\varphi(\tilde{X}_i, \tilde{X}_j) := \sqrt{\mathbb{E}[\tilde{X}_i^2]\mathbb{E}[\tilde{X}_j^2]} - \mathbb{E}[\tilde{X}_i \tilde{X}_j]$$

$$= \frac{1}{T}\|\tilde{X}_i\|\|\tilde{X}_j\| - \frac{1}{T}\tilde{X}_i^T \tilde{X}_j$$

$$= \frac{1}{T}\|\tilde{X}_i\|\|\tilde{X}_j\|\{1 - CosSim(\tilde{X}_i, \tilde{X}_j)\}$$

- Desired properties:

1. Insensitive to the length of document
2. Sensitive to the magnitude of the signal
3. Increasing function of cosine distance

Using Pre-Trained Word Embeddings

We utilize some widely used word embeddings:

- **GloVe**[5] pre-trained on Wikipedia 2014 and Gigaword 5 with vocabulary size of 400K and 300d vectors
- **fastText**[6] pre-trained on Wikipedia 2017 with the vocabulary size of 1M and 300d vectors
- **ConceptNet Numberbatch** [7] *v*17.06 with the vocabulary size of 400K and 300d vectors

[5] [Pennington et al., 2014]
[6] [Bojanowski et al., 2016, Joulin et al., 2016]
[7] [Speer et al., 2017]

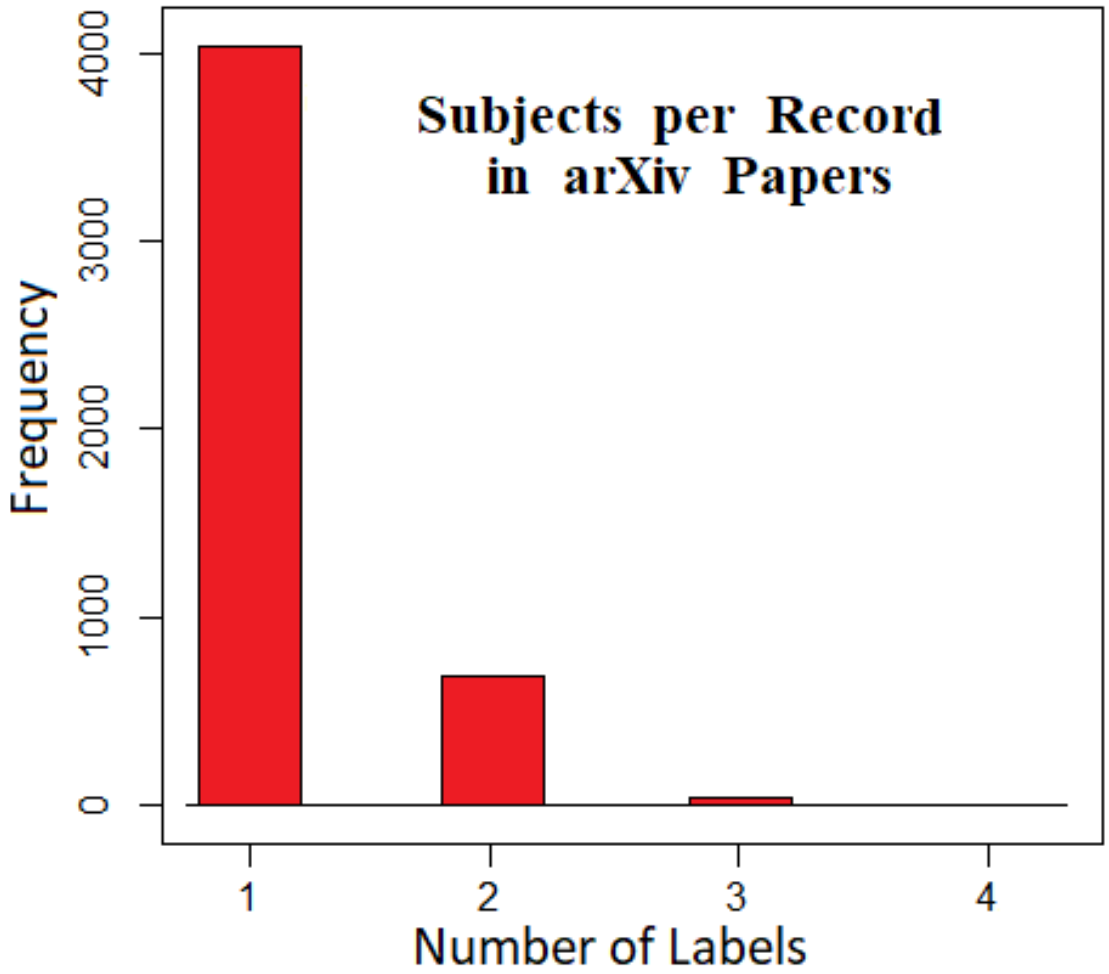
Data to Examine TDA on Word Embeddings

Date Specification for arXiv papers and IMDB reviews.

Specification	arXiv Quant. Fin. Papers	IMDB Movie Reviews
Labels	5 (Multi-label)	2
Clean Records	4601	6000
Length of Records	8456.9 ± 6395.8	540.5 ± 171.9
Frequency of Labels	*q-fin.GN* : 1258 *q-fin.ST* : 1144 *q-fin.MF* : 977 *q-fin.PR* : 907 *q-fin.RM* : 913	*Positive* : 3000 *Negative* : 3000

Data to Examine TDA on Word Embeddings

Histograms of number of labels per document in arXiv data set of papers.



Results using TDA on Word Embeddings

On **arXiv papers** dataset, TDA achieves better F1 and Acc.

Model	Embedding	Window	Prec.	Rec.	F1	Acc.
Topology + XGBoost	fastText	3	61.9	55.4	0.575	80.1
Topology + XGBoost	GloVe	3	63.1	56.7	0.597	80.7
Topology + XGBoost	Numberbatch	3	**68.7**	**60.5**	**0.643**	**82.6**
Topology + XGBoost	fastText	5	60.8	54.7	0.576	79.8
Topology + XGBoost	GloVe	5	61.8	56.1	0.588	80.3
Topology + XGBoost	Numberbatch	5	65.5	58.4	0.617	81.6
Topology + XGBoost	fastText	7	58.9	54.4	0.566	79.5
Topology + XGBoost	GloVe	7	62.8	56.4	0.594	80.6
Topology + XGBoost	Numberbatch	7	65.7	57.7	0.614	81.3
Topology + XGBoost	fastText	7 expon.	60.3	54.6	0.573	79.7
Topology + XGBoost	GloVe	7 expon.	61.2	55.9	0.584	80.2
Topology + XGBoost	Numberbatch	7 expon.	66.4	59.6	0.628	82.2
CNN	fastText	-	57.1	64.3	0.605	80.0
CNN	GloVe	-	57.6	64.2	0.607	80.6
CNN	Numberbatch	-	**55.0**	**67.6**	**0.607**	**79.8**

Results using TDA on Word Embeddings

On **IMDB Reviews** dataset, TDA achieves slightly better F1 and Acc.

Model	Embedding	Window	Prec.	Rec.	F1	Acc.
Topology + XGBoost	fastText	3	84.8	85.8	0.853	85.4
Topology + XGBoost	GloVe	3	86.9	88.0	0.874	87.5
Topology + XGBoost	Numberbatch	3	**87.9**	**89.0**	**0.884**	**88.5**
Topology + XGBoost	fastText	5	84.2	85.2	0.847	84.8
Topology + XGBoost	GloVe	5	85.6	86.6	0.861	86.2
Topology + XGBoost	Numberbatch	5	86.5	87.6	0.870	87.1
Topology + XGBoost	fastText	7	82.8	83.8	0.833	83.4
Topology + XGBoost	GloVe	7	83.8	84.8	0.843	84.4
Topology + XGBoost	Numberbatch	7	85.3	86.3	0.858	85.9
Topology + XGBoost	fastText	7 expon.	84.3	85.3	0.848	84.9
Topology + XGBoost	GloVe	7 expon.	86.5	87.6	0.870	87.1
Topology + XGBoost	Numberbatch	7 expon.	87.0	88.1	0.875	87.6
[Shaukat et al., 2020]	Lexicon based	-				86.7
[Giatsoglou et al., 2017]	Hybrid (TF/IDF + Embeddings)	-			**0.880**	**87.8**

Results using TDA on Word Embeddings

Results per class on arXiv papers dataset using ConceptNet Numberbatch as pre-trained embedding and window size of 3.

Subject	Test Records	Precision	Recall	F1	Accuracy
q-fin.GN	410	73.2	68.5	**0.708**	83.8
q-fin.ST	396	70.2	67.5	0.688	83.6
q-fin.MF	306	66.0	45.6	**0.539**	77.5
q-fin.PR	305	69.5	55.2	0.615	82.7
q-fin.RM	307	62.5	61.0	0.617	84.5

The best F1 achieved for "General" class, while the worst case is for "Mathematical Finance" probably because the set of its class-specific terms has many intersections with other classes.

Data to Examine TDA on TF-IDF & Word Embeddings

Number of records per class and overlaps

Specification	Drama	Comedy	Action	Romance
Overlap with drama	-	524	223	379
Overlap with comedy	524	-	207	544
Overlap with action	223	207	-	117
Overlap with romance	379	544	117	-
Exclusive Records	4592	3302	1181	672
Total Records	5615	4477	1658	1614

Results using TDA on TF-IDF & Word Embeddings

Our ensemble models provide the best results for every genre.

Accuracy per class using different method

Class	BiLSTM	XGB	XGB2	LR	LR2	prev.SVC	prev.NB
action	87.7	86.7	86.9	**89.3**	88.9	81.5	82.7
comedy	75.6	69.0	69.1	76.9	**77.7**	74.6	73.3
drama	69.9	63.9	64.3	71.0	**71.6**	66.1	67.4
romance	87.6	86.0	85.9	**87.8**	**87.8**	88.3	84.3
macro-avg	76.2	71.1	71.4	77.3	**77.8**	73.5	73.3

- XGB: XGBoost using TP1 (Topological features of word embeddings)
- XGB2: XGBoost using both topological feature sets.
- LR: Logistic regression combining the results of XGB and BiLSTM
- LR2 : Logistic regression combining the results of XGB2 and BiLSTM
- prev SVC: previous results using linear SVC
- prev NB: previous results using multinomial Naïve Bayes.

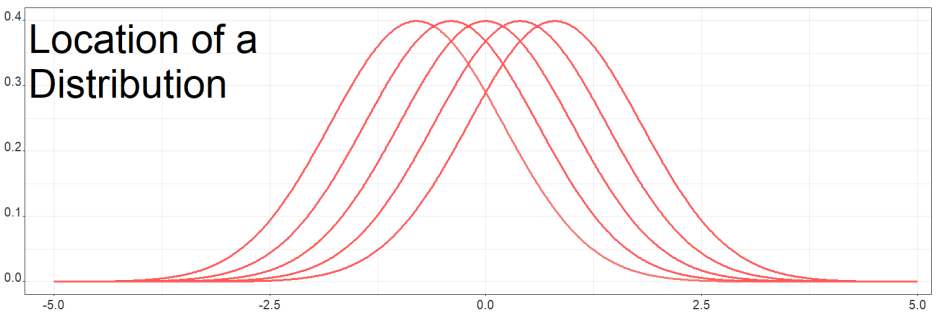
Results using TDA on TF-IDF & Word Embeddings

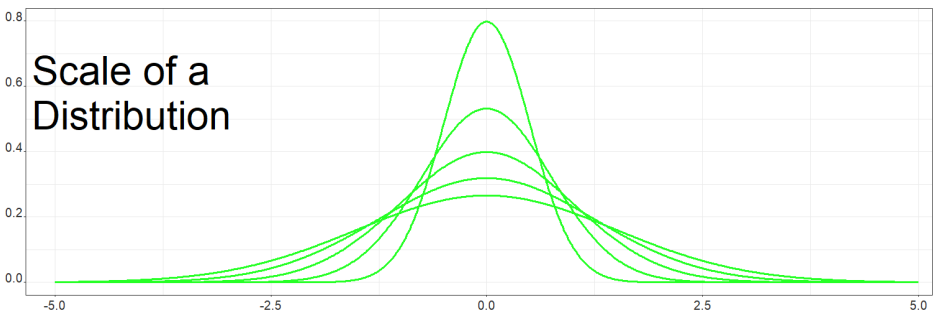
- Topological features from word embedding space can classify the records alone with an accuracy comparable but not equal to the LSTM.
- Topological features extracted from TF-IDF space are primarily used to reflect some repetitive patterns in the text.
- Using the topological feature sets can boost the accuracy of classification in the ensemble model.

TDA Using TF-IDF: Details of the Algorithm

Table: Features from dimension 1 extracted from TF-IDF representation.

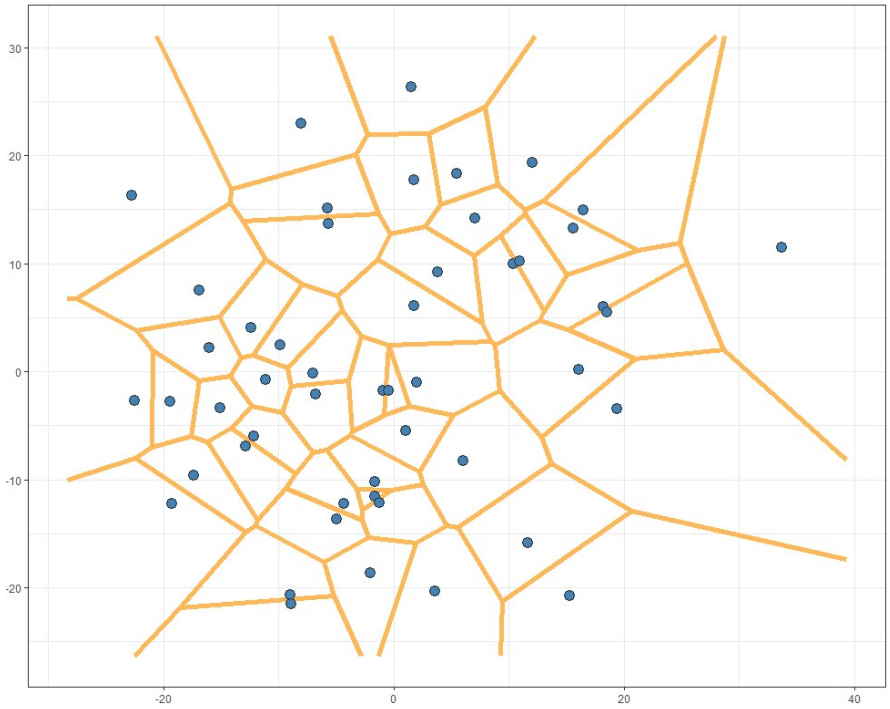
	Feature	Loops Info.	Prev. Applications
1	Number of loops	Loops variety	[Zhu, 2013] to find repetitive text patterns
2	Avg diameter of birth	Birth location	
4	Avg duration	Duration location	[Mittal and Gupta, 2017] for dynamical systems
3	Std Dev of birth diameters	Birth scale	
5	Std Dev of durations	Duration scale	





Introduction: Topology vs. Geometry

- Voronoi Diagram: partitioning the space to convex sub-regions.
- Geometry depends on distances.
- Geometry
  $\Rightarrow$ Voronoi diagram
  $\Rightarrow$ Distance-based models in machine learning
- TDA techniques utilize:
  - similar intuition
  - more complex methodologies



- Topology is robust under stretching or shrinking.
- TDA methods are much less sensitive to the choice of metric.

Can AI Consider Complicated Scenarios?