Persistent Homology in Text Mining

ACAT Meeting, Bremen

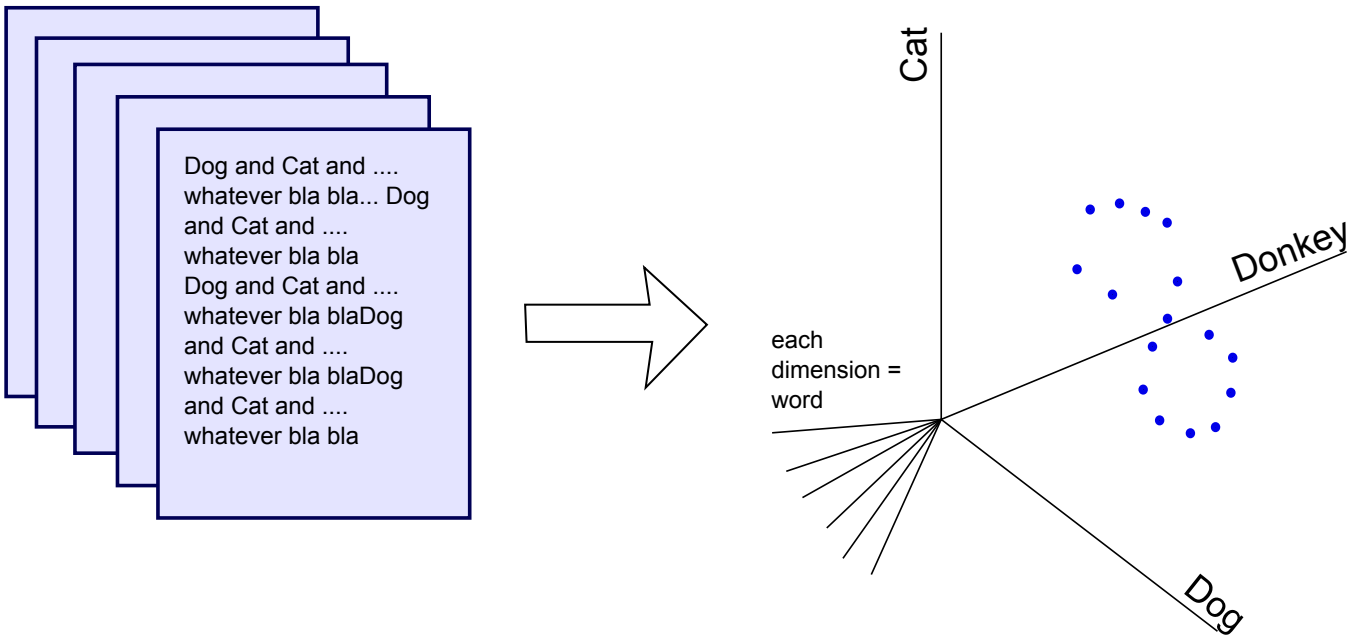
Hubert Wagner (Jagiellonian University)

Joint work with Pawel Dlotko (UPenn) and Marian Mrozek (Jagiellonian University)

July 18, 2013

Big picture.

- Topology can be useful in analyzing text data.
- We 'sold' this idea to Google.
- The input is local: documents and their *similarities*.
- Persistence gives some geometrical-topological global information, describing the entire *corpus* (set of documents).



Plan

- Text data and its representation.
- Concepts from text mining, similarity measure.
- Extended similarity measure.
- Practical usage.

Practical example of text mining application

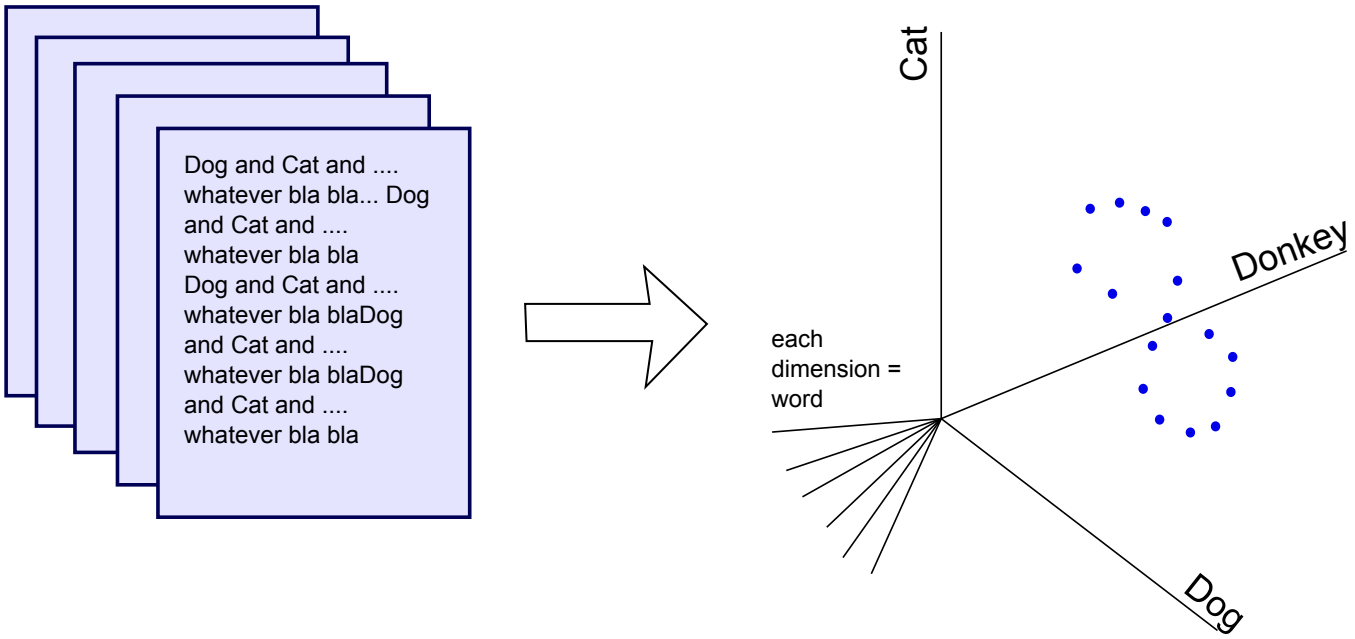
- Google Alerts (www.google.com/alerts)
- 'Monitor the Web for interesting new content'.
- You specify the query (topic, keywords).
- It 'googles' the given topic every day for you.
- Email notification when something *new* appears.
- Problem: lots of spam: most results people got pointed to very *similar* webpages/documents.

Interesting properties of text data.

- Zipf law, intuitively: relative frequency of the k-th most popular word is roughly 1/k.
- For a reasonable corpus:
  - k = 1 gives 6% (in English: 'the')
  - k = 2 gives 3% ('of')
  - k = 3 gives 2% ('to')
- It works for all natural languages...

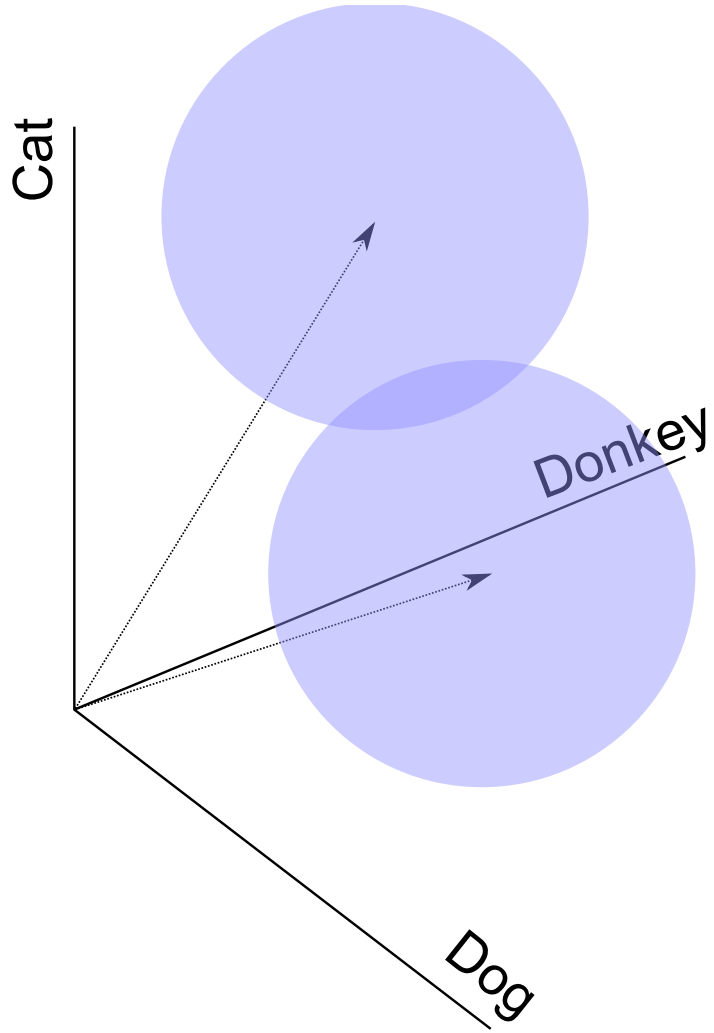
Representation of text data.

Each point represents a single document, ideally its position summarizes the content/topic.



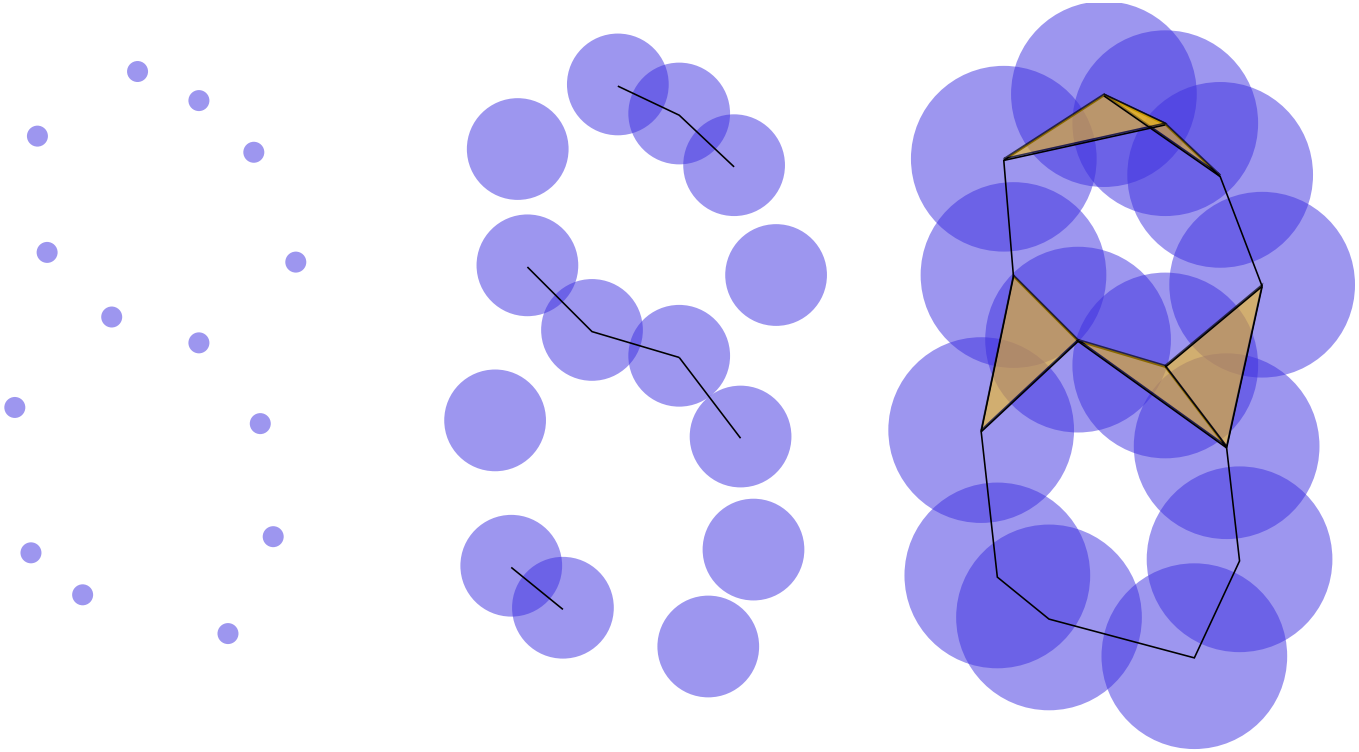
Representation of text data.

- It's natural to think about similarity between text documents.
- 'Balls' wrt. similarity describe a *context* (for some radius).



Shape of such data.

- With persistence we can view it at different scales, namely: similarity thresholds.
- Each simplex means that its vertices/documents are similar (at this similarity scale/threshold).



Existing tools.

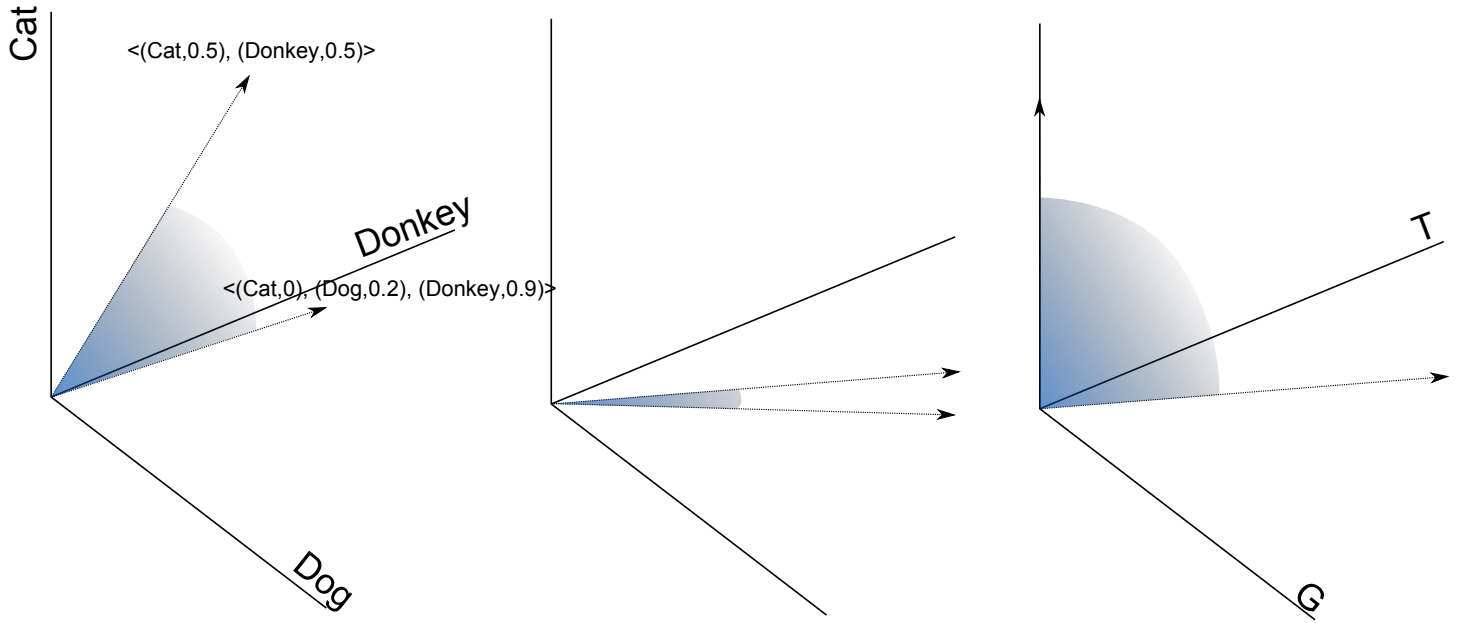
- There are text-mining techniques to represent the data (Vector-Space-Model).
- A standard similarity measure which works well in practice (cosine similarity).
- Of course we can just build a Rips complex...
- Is this the right method?

Concept: Term-vectors

- Used to extract characteristic words (or *terms*) from a document.
- Each term is weighted according to its relative 'importance'.
  - Words which appear often in a document are weighted higher. But this is offset by their global frequency. ('tf-idf')
- Usually at most 50 non-zero coefficients.
- So, each document is described by vector of pairs: $(term_i, weight_i)$, only non-zero weight matters.

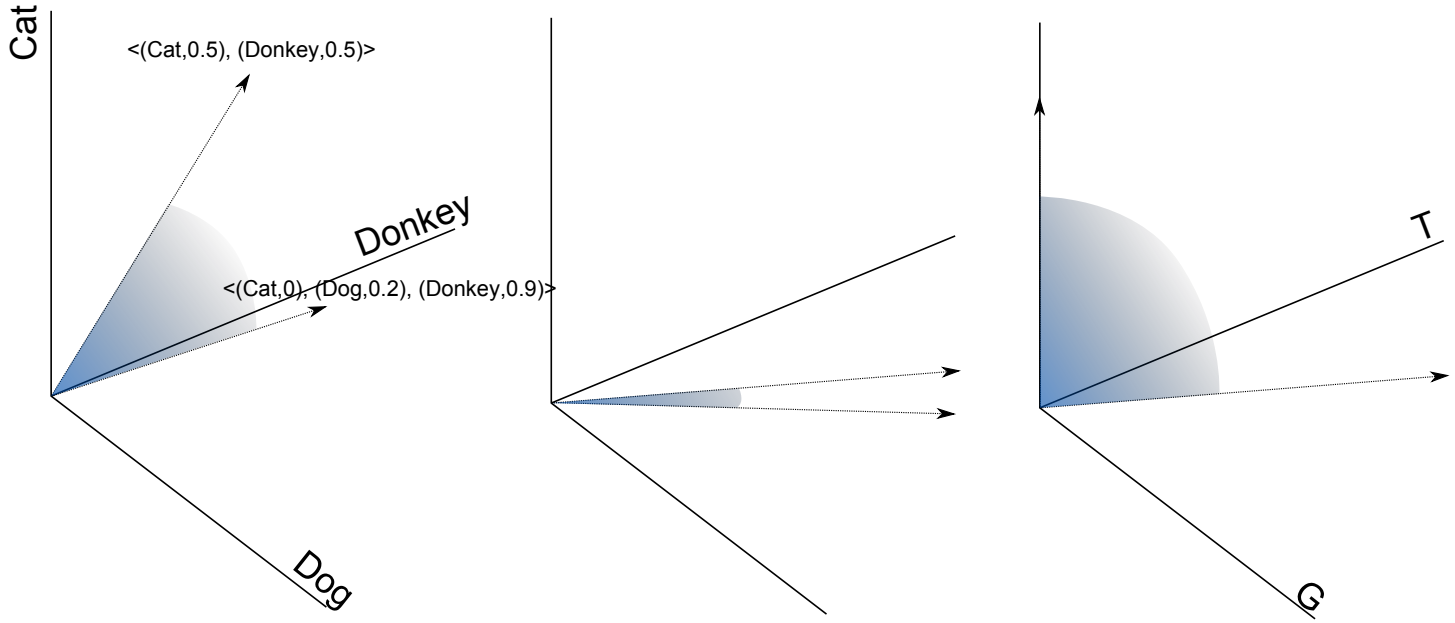
Concept: Vector Space Model

- *Vector Space Model* maps a corpus to $\mathbb{R}^d$.
- Each document is represented by its term-vector.
- Each unique term becomes an orthogonal direction, so the (embedding) dimension d can be very high.
- Term-vectors give the coordinates of documents in this space.
- It was a huge breakthrough in the 80s for information retrieval, text mining etc. Still used!



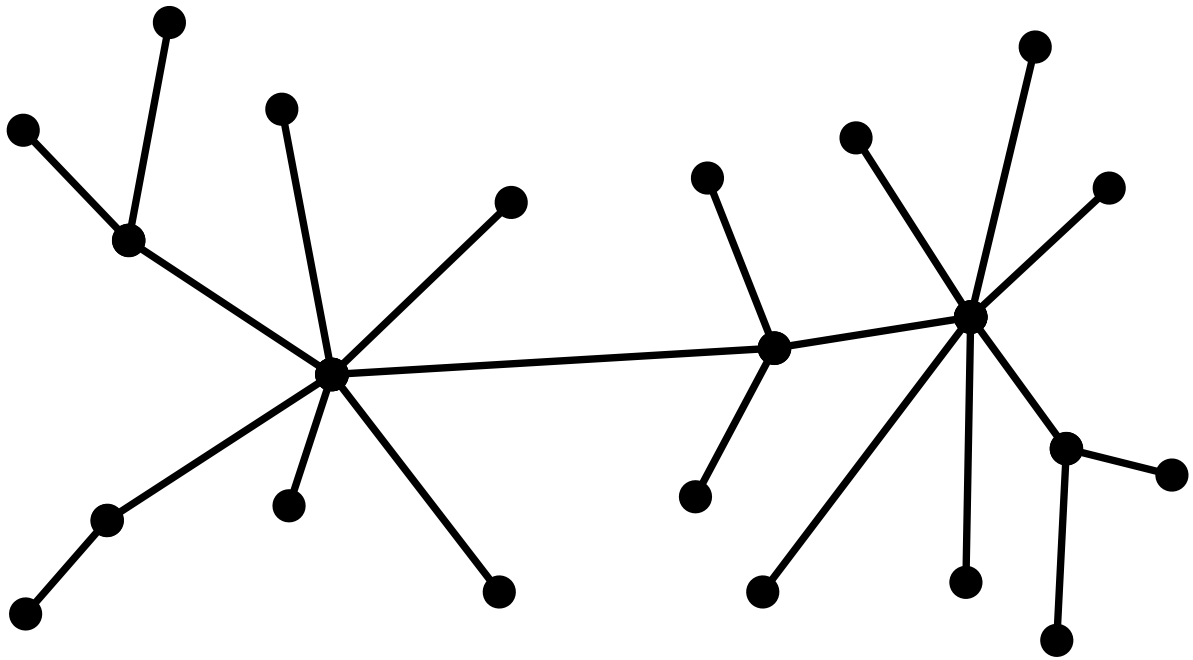
Concept: Vector Space Model and similarity

- We use the *cosine similarity* to 'compare' two documents.
- 'Dissimilarity': $dsim(a, b) := 1 - sim(a, b)$.
- Dissimilarity is not a metric (no triangle inequality).



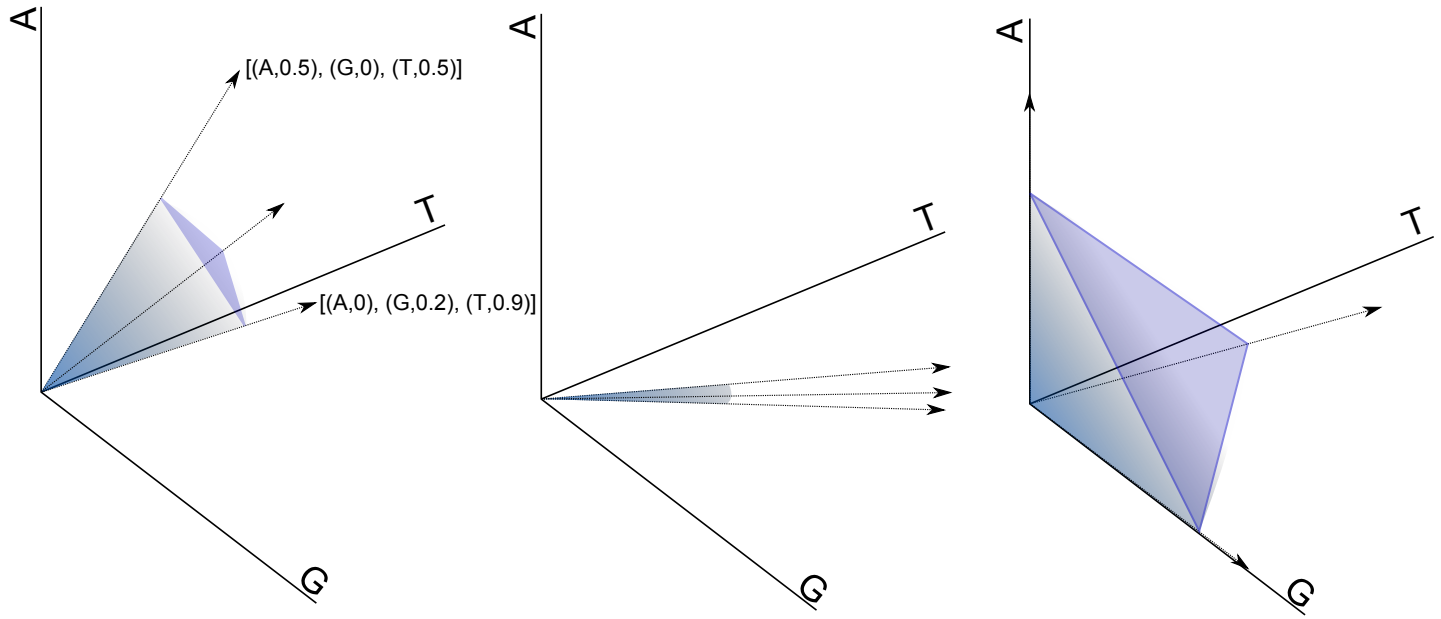
Another interesting property.

- Even 0-homology is interesting.
- The neighborhood graph has a special structure.
- It's a scale-free graph.
- Random graph model: preferential attachment (Barabási-Albert).
- Hyperlinks, social networks, citation networks also follow this.



New concept: Extended Vector Space Model and multidimensional similarity

- We extend the notion of similarity from pairs to larger subsets of documents (up to size, say, 5).
- This way we capture 'higher-dimensional' relationships in the input.
- The resulting simplicial complex is a Cech complex.



Concept: Extended Vector Space Model and multidimensional similarity

- $Sim(X_1..X_d) = \frac{\sum_i \prod_{j=1}^{d} X_{ij}}{\prod_{i=1}^{d} \|X_i\|_d}$. For $d = 2$ it's the cosine.
- Extends similarity from pairs to larger subsets of documents.
- The value of each d-simplex gets the similarity among the $q + 1$-subset of documents it contains.

cat	dog	donkey
0.8	0.5	0
0	0.8	0.4
0.3	0	0.7

Concept: Extended Vector Space Model and multidimensional similarity

- $Sim(X_1..X_d) = \frac{\sum_i \prod_{j=1}^{d} x_{ij}}{\prod_{i=1}^{d} \|x_i\|_d}$.
- Intuition: for binary weights, Sim is the size of set-theoretical intersection (up to normalization). For $d = 2$ it's (almost) the Jaccard measure.

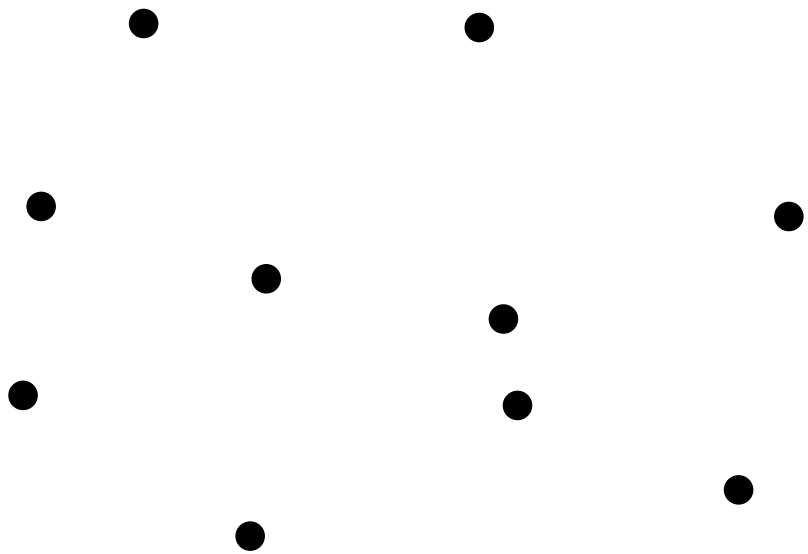
cat	dog	donkey
1	1	0
0	1	1
1	0	1

Our experimental setting.

- We use documents from the English Wikipedia.
- Input: point cloud $P \subset \mathbb{R}^d$
- Build the Cech complex, filtered by dissimilarity.
- Remember: each simplex gets filtration value = dissimilarity of its documents.
- Compute and analyze persistence diagrams.

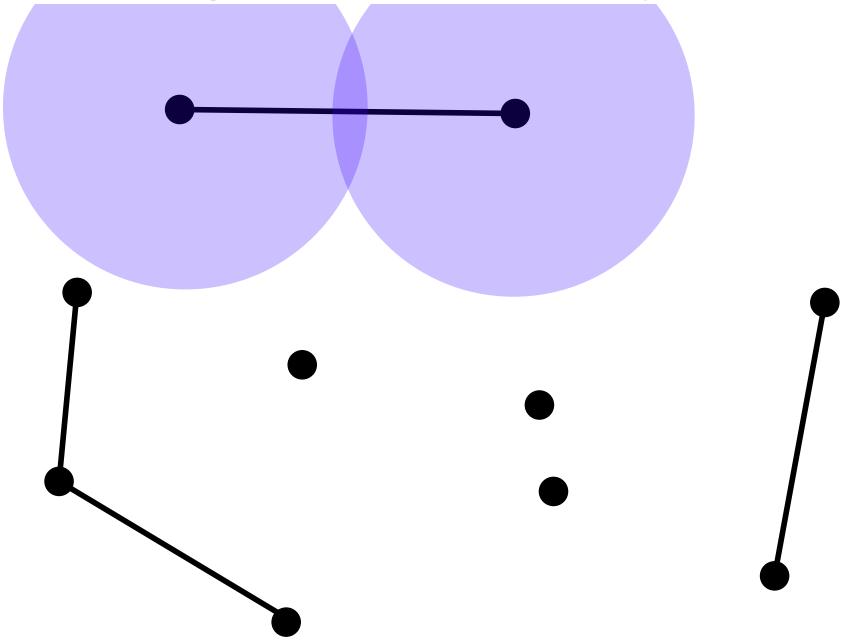
Persistence of these data.

- We increase our dissimilarity threshold...
- ... allowing less and less related documents to be considered 'similar'.
- We check how holes are created and how they merge (the younger one 'dies') during this process...
- We change the dissimilarity threshold from 0 to 1:



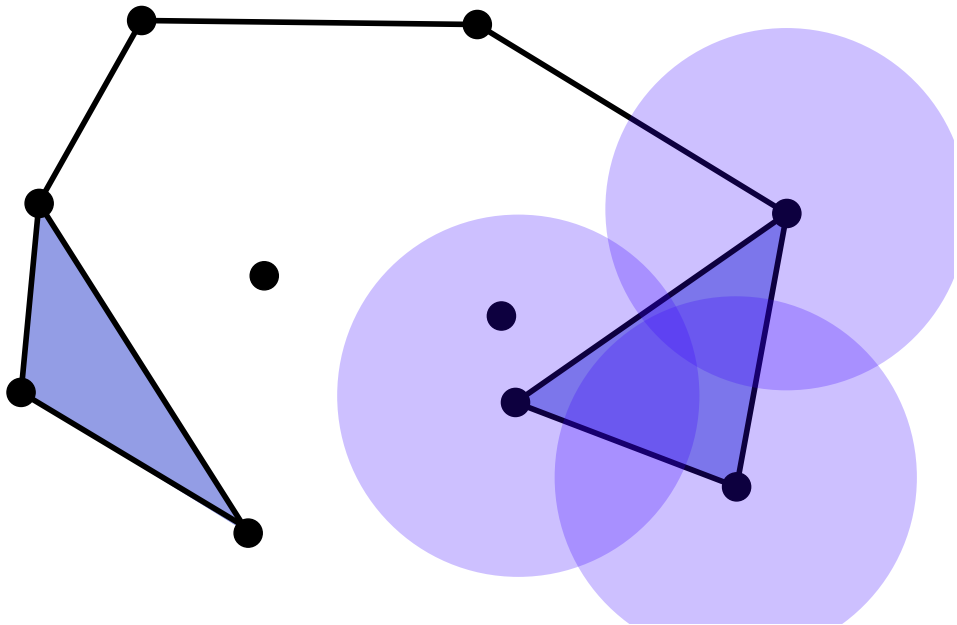
Persistence of these data.

- We increase our dissimilarity threshold...
- ... allowing less and less related documents to be considered 'similar'.
- We check how holes are created and how they merge (the younger one 'dies') during this process...
- We change the dissimilarity threshold from 0 to 1:



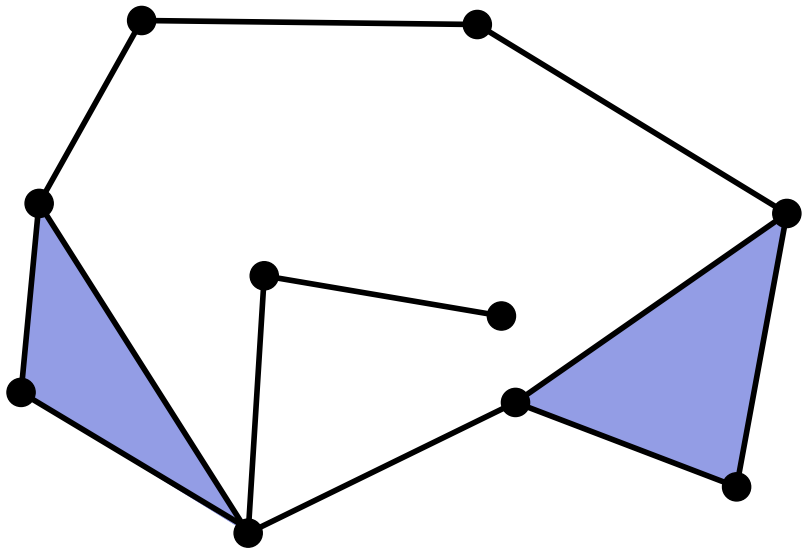
Persistence of these data.

- We increase our dissimilarity threshold...
- ... allowing less and less related documents to be considered 'similar'.
- We check how holes are created and how they merge (the younger one 'dies') during this process...
- We change the dissimilarity threshold from 0 to 1:



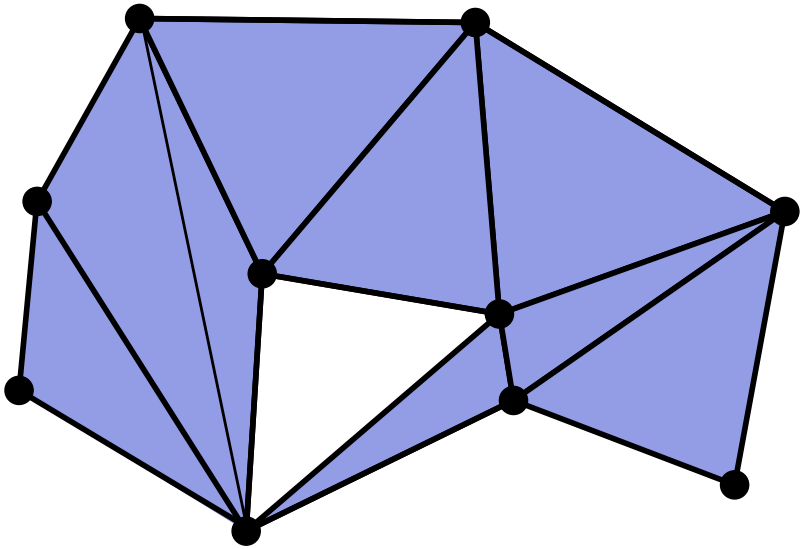
Persistence of these data.

- We increase our dissimilarity threshold...
- ... allowing less and less related documents to be considered 'similar'.
- We check how holes are created and how they merge (the younger one 'dies') during this process...
- We change the dissimilarity threshold from 0 to 1:



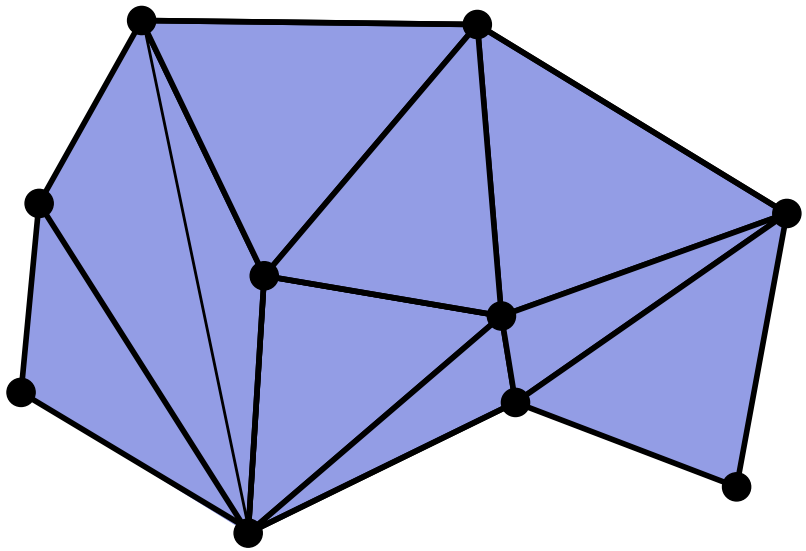
Persistence of these data.

- We increase our dissimilarity threshold...
- ... allowing less and less related documents to be considered 'similar'.
- We check how holes are created and how they merge (the younger one 'dies') during this process...
- We change the dissimilarity threshold from 0 to 1:



Persistence of these data.

- We increase our dissimilarity threshold...
- ... allowing less and less related documents to be considered 'similar'.
- We check how holes are created and how they merge (the younger one 'dies') during this process...
- We change the dissimilarity threshold from 0 to 1:



Big picture again.

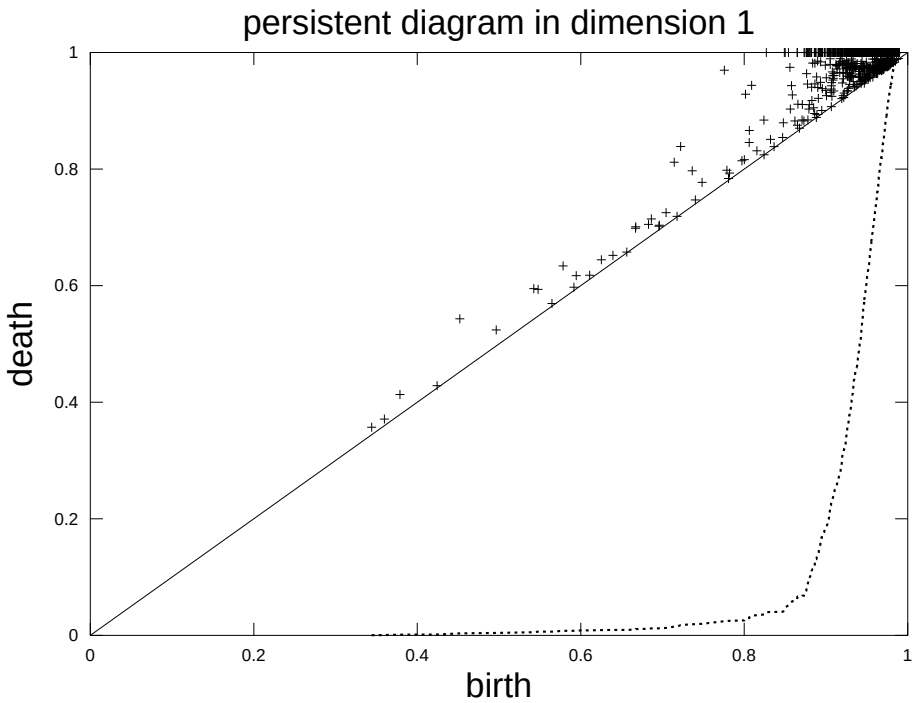
- We are interested in 'topology' of textual data in this representation.
- More precisely: in the global structure of similarities among documents.
- We can capture high-dimensional relationships (extended similarity).
- Overall it gives (some) global information of the *entire corpus*.

Applications

- Dimensionality estimation.
- Interactive text data exploration, attention routing, missing data.
- Inter-language comparison of corpora, stability.
- Simplification (overview) of data.

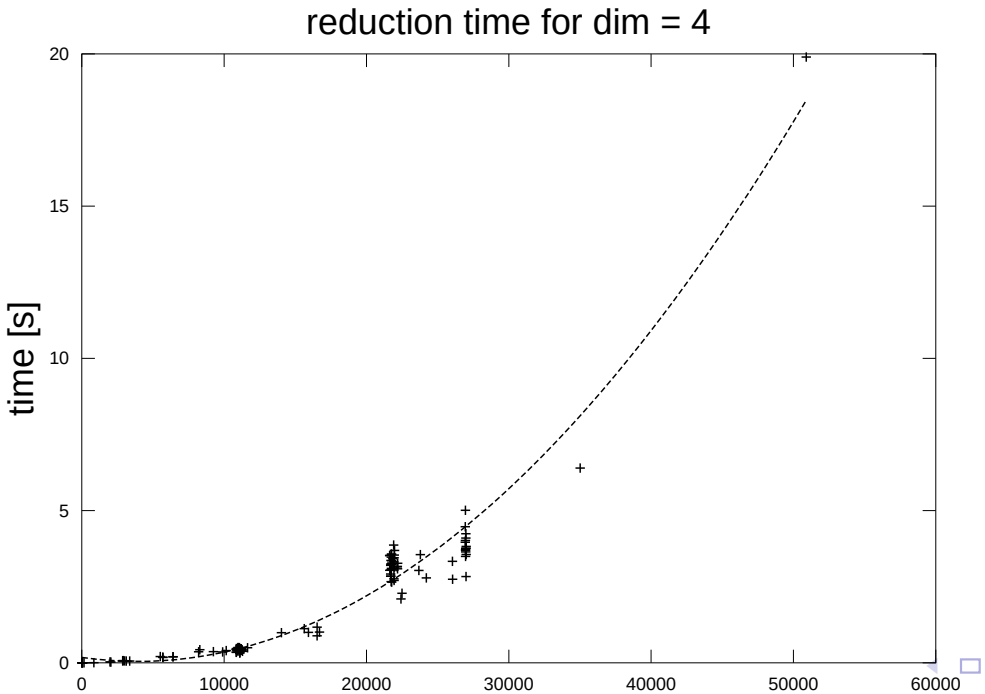
Persistence in dim 1.

We see some phase transition around dissimilarity value $= 0.8$.



Computational results: "Can you do this for 10^{14} points?"

- In practice the number of simplices is at least 10^9.
- Standard method to compute persistence: reduce the ordered boundary matrix.
- Efficiency is a problem for such datasets (quadratic scaling, worst-case is cubic).
- We want a general *tool* which to do experiments and research with...



Practical efficiency

For text-data using Rips complexes and standard computational methods we experienced quadratic running time (in the size of the complex) and the complexes were large. We now see *linear* running times and significantly smaller complexes, by:

- Switching to persistent cohomology (duality due to Vin de Silva et al.).
- Cech complex is significantly smaller.
- No need for preprocessing: all pairs have non-zero persistence.
- Using the new efficient PHAT library (Ulrich Bauer, Michael Kerber, Jan Reininghaus).
- Additionally new types of simplicial complexes look promising (Graph-induced, Zig-zag zoos...)

Conclusion

- New setup, capturing higher-dimensional relationships.
- We can construct a Cech complex, which is smaller than Rips.
- With new tools, we can handle reasonably-sized text data.
- We hope to use it to answer questions for some real-world data.

- Thanks for Ulrich Bauer, Herbert Edelsbrunner, Jan Reininghaus for helpful comments.
- Some of these is published: HW, P.Dlotko, M.Mrozek, "Computational Topology for Text Mining", CTIC 2012.
- Also: you can check out the persistence library: code.google.com/p/phat/
- Thank you!
- (Research supported by UE Programme: "Geometry and Topology in Physical Models" and Google Research Award programme.).