The Hidden Potential of non-Euclidean Data Representations for Improved Machine Learning

Nakul Verma

Columbia University

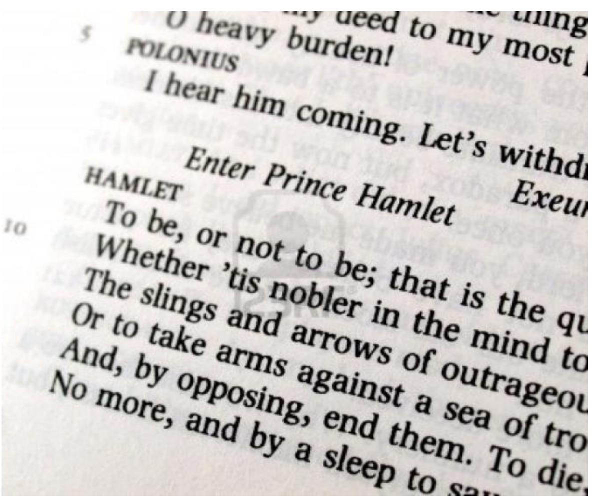
Nature of modern data

Several real-world data are naturally represented as collection of points in very high dimensions

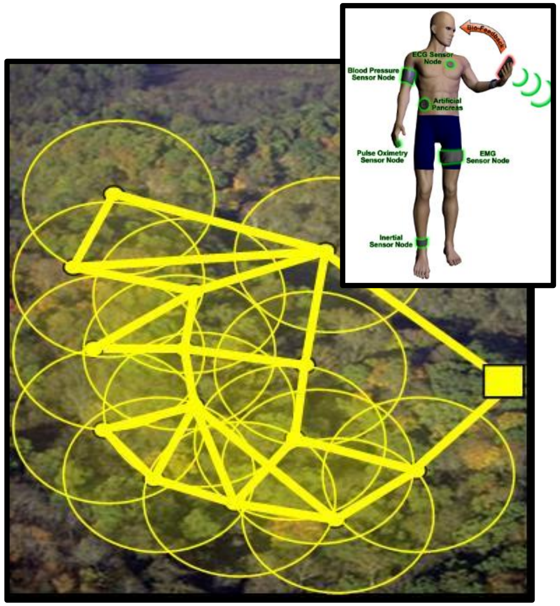
Some Examples

Images

Text



Sensor Networks



Fundamental question

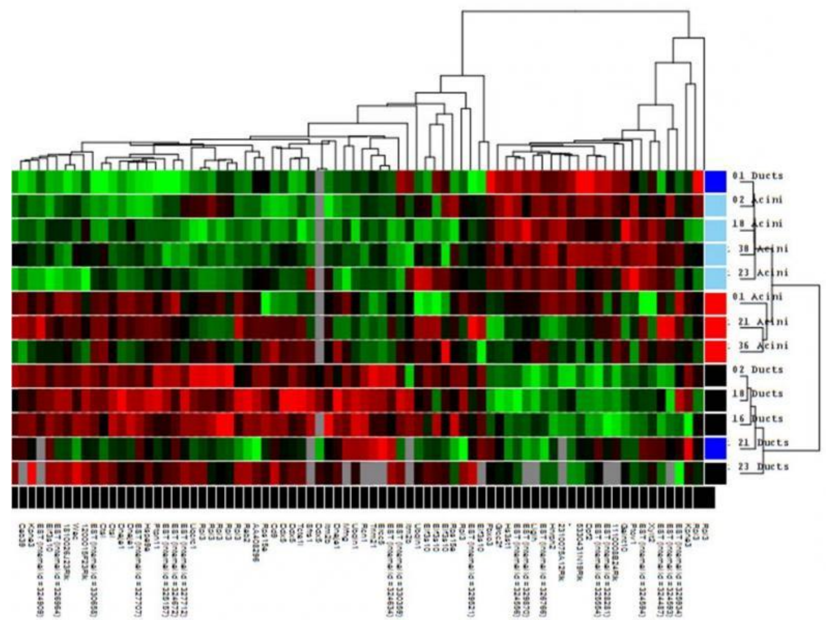
How can we design effective learning algorithm for data that is represented in very high dimensions?

Key Observation:

Even though data is represented in high dimensions, it typically conform to some intrinsic structure

Examples of intrinsic structure

- manifold structure
- sparse structure
- taxonomy structure



Nature of modern data

As a practitioner one hopes:

With **richer representation** of data, we have more information; and thus should be able to guarantee that specific ML tasks such as prediction, classification, recommendation better.

Unfortunately…

Bad News…

Having inappropriate data representation often leads to

- Poor understanding of global relationships in data
- Wastefully large dimensionality (leading to curse of dimensionality)
- Complicated learning models
- Low quality predictions

We spend time on designing good, finely tuned ML models,

BUT often forget to think about effective and appropriate data representations

Outline of the Talk

- **Topological analysis of data and the need for non-Euclidean representations**

- **Case study: Hyperbolic representations of hierarchical data**

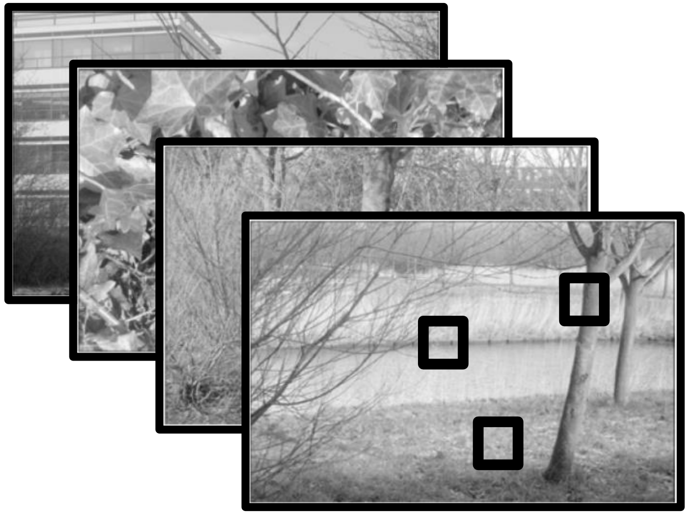
- **Technical considerations when working with non-Euclidean representations.**

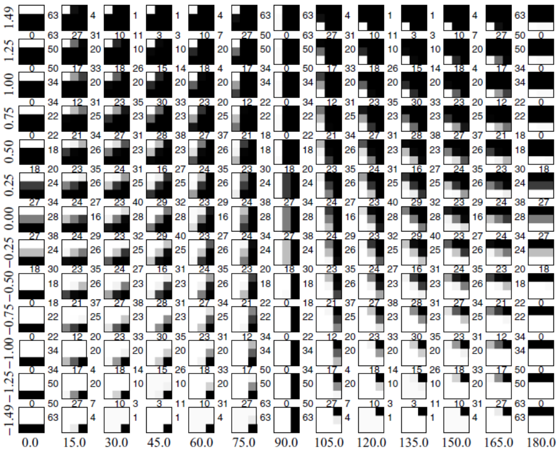
- **Improved prediction quality in non-Euclidean representations**

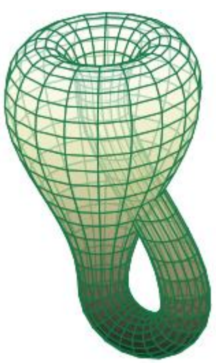
- **Future directions, and further discussion**

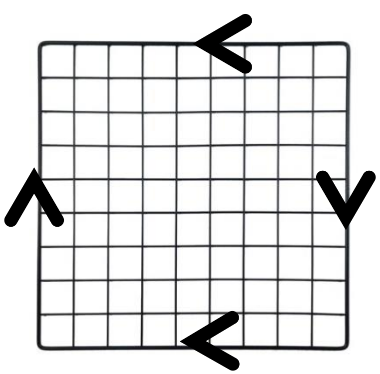
A closer look at data

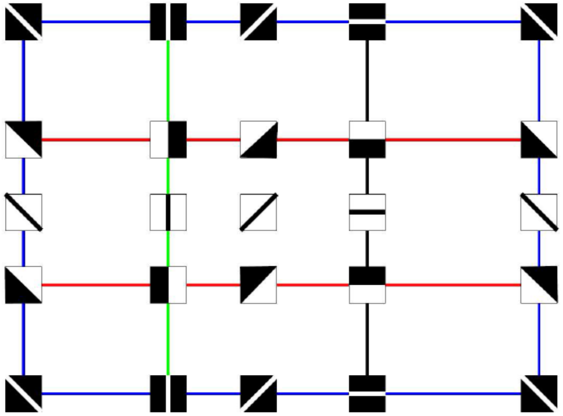
Natural image patches have a Klein Bottle topology







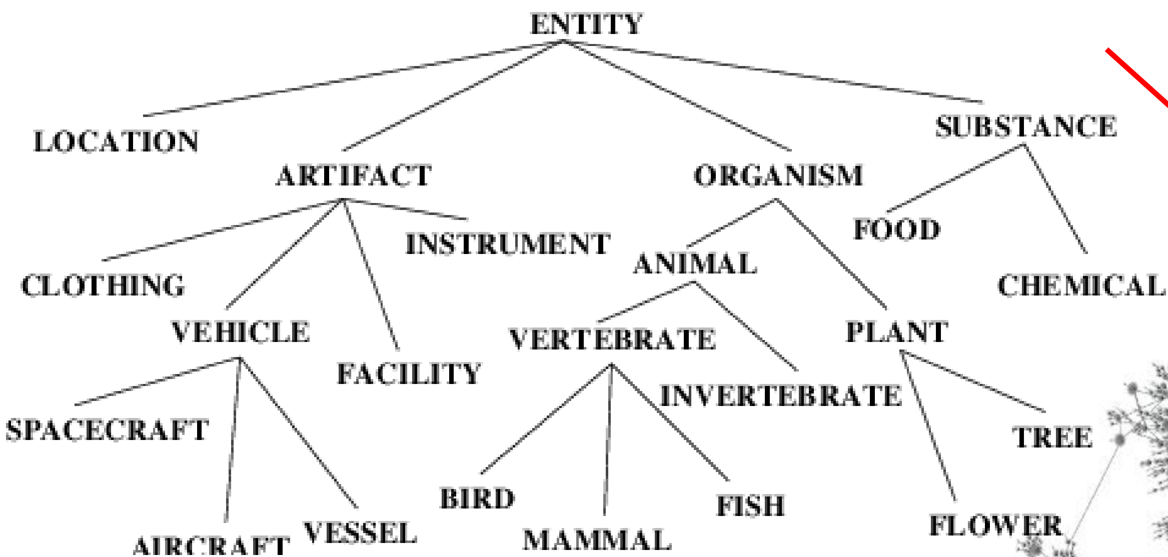


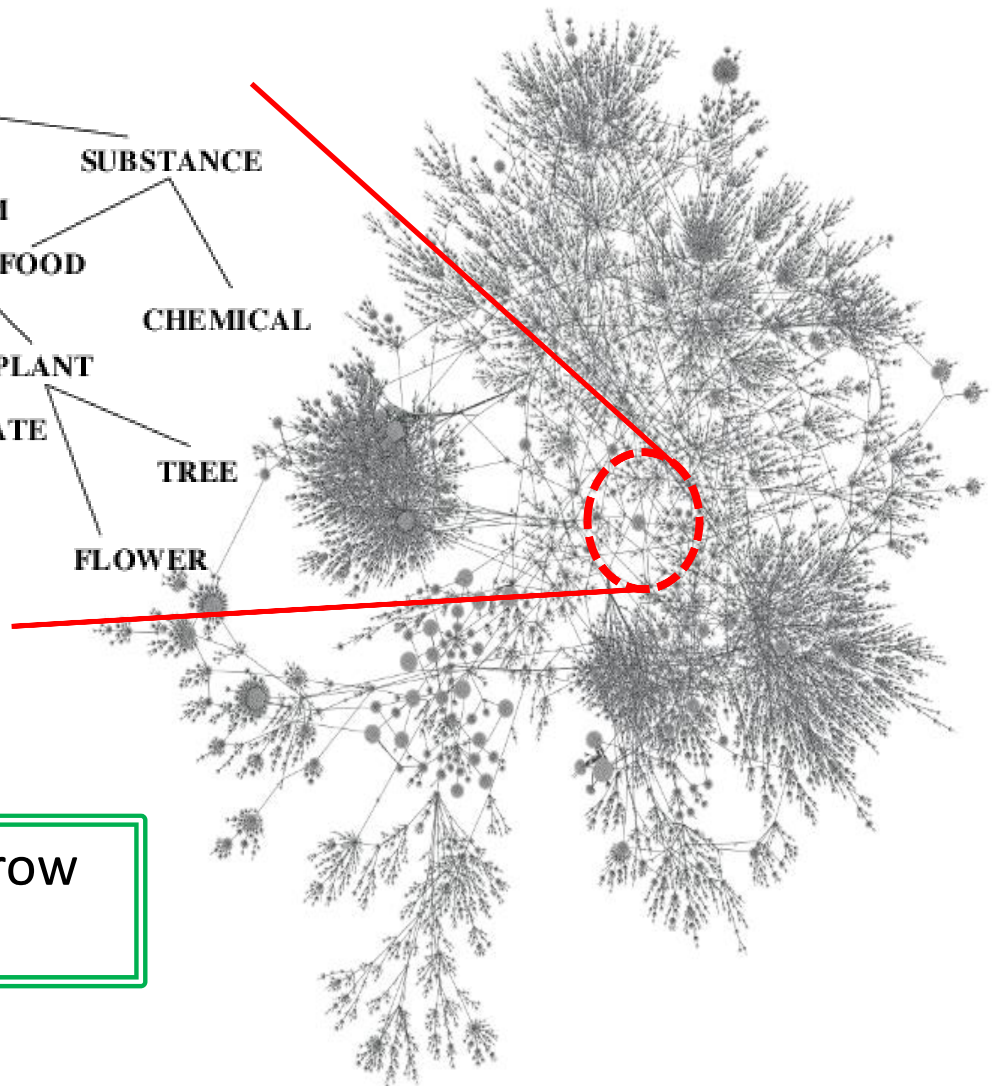


[Carlsson et al., 2008]

A closer look at data

Hierarchical data is naturally represented in hyperbolic spaces





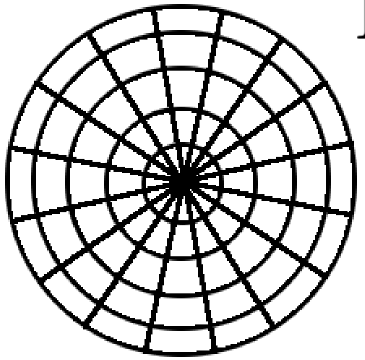
Observation: number of leaves grow exponentially with tree depth

A closer look at data

Observations:

1. Number of leaves grow exponentially with tree depth.
2. Unfortunately, *d*-dimensional surface of a ball in Euclidean space grows only polynomially with radius.

$\mathbb{R}^2$

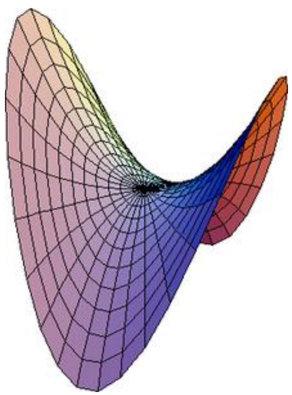


Consequence: Euclidean space ***cannot*** adequately represent hierarchical data!

Need a representation space that grows exponentially fast with distance…

eg. Hyperbolic space!

$\mathbb{H}^2$



Of Hierarchies and Hyperbolic spaces

Since hierarchical data is prevalent, data analysis on hyperbolic spaces is gaining much attention.

Difficulties hyperbolic spaces:

- Cannot even do basic operations like vector addition!

There are now hyperbolic ML models for… [2017-present]

- Multi dimensional Scaling (MDS)
- Support Vector Machines (SVMs)
- Recommender systems (RecSys)
- Neural networks (NN)

Algorithm design relies crucially on the structure of hyperbolic spaces and **cannot** extend to other exotic spaces.

ML on non-Euclidean spaces

Can we re-design ML algorithms that can work in generic non-Euclidean spaces?

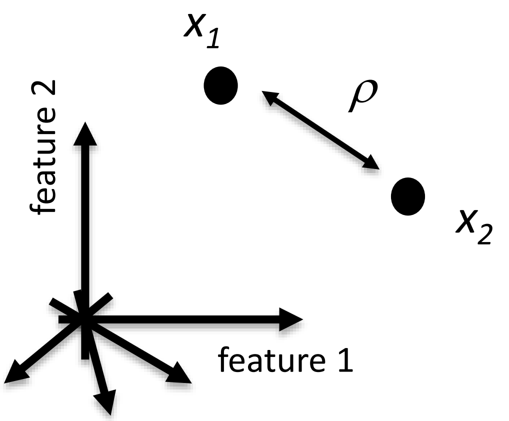
Some classic machine learning algorithms **can be extended** to generic non-Euclidean spaces (beyond, Hyperbolic spaces)

- Metric Learning
- *k*-means clustering
- Multi-dimensional Scaling

[Aalto and Verma, 2019]

Metric Learning in Euclidean Spaces

Comparing observations in feature space:



$$\rho(x_1, x_2) = \|x_1 - x_2\|^2$$
$$= (x_1 - x_2)^\mathsf{T}(x_1 - x_2)$$

[*sq.* ***Euclidean*** *dist*]

(all features are equally weighted)

$$\rho_M(x_1, x_2) = \|M(x_1 - x_2)\|^2$$

(using weighting mechanism M)

$$= (x_1 - x_2)^\mathsf{T}(M^\mathsf{T} M)(x_1 - x_2)$$

[*sq.* ***Mahalanobis*** *dist*]

Q: What should be the correct weighting *M*?

A: Problem is dependent and data-driven.

Given data of interest, *learn* a *metric* (*M*), which helps in the prediction task.

How to Learn Optimal Weighting?

Want:

Distance metric: $\rho_M(\vec{x}_1, \vec{x}_2)$

such that: data samples from **same class** yield **small values**

data samples from different class yield large values

One way to solve it mathematically:

Create **two** sets: Similar set $S := \{(\vec{x}_i, \vec{x}_j) \mid y_i = y_j\}$

Dissimilar set $D := \{(\vec{x}_i, \vec{x}_j) \mid y_i \neq y_j\}$

i, *j* = 1,..., *n*

Define a cost function:

$$\Psi(M) := \lambda \sum_{(\vec{x}_i, \vec{x}_j) \in S} \rho_M(\vec{x}_i, \vec{x}_j) \; - \; (1-\lambda) \sum_{(\vec{x}_i, \vec{x}_j) \in D} \rho_M(\vec{x}_i, \vec{x}_j)$$

Minimize Ψ w.r.t. M!

Empirical performance (faces dataset)

Query

learned metric

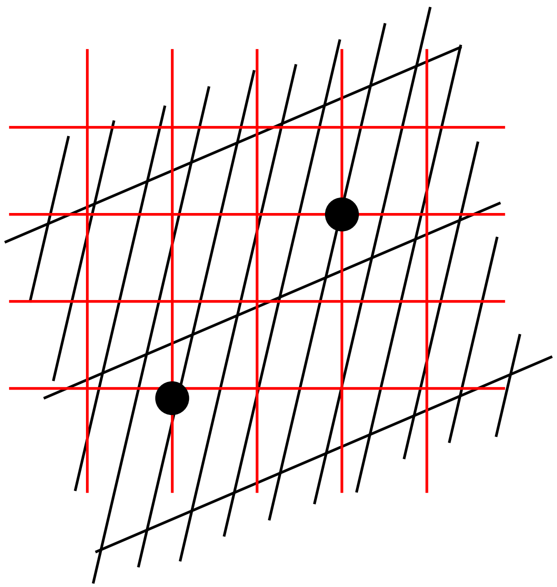
Original space

[Weinberger and Saul, 2009]

Metric Learning in non-Euclidean Spaces

Observation:

Reweighting of features via Metric Learning can be thought as transforming the underlying coordinate system

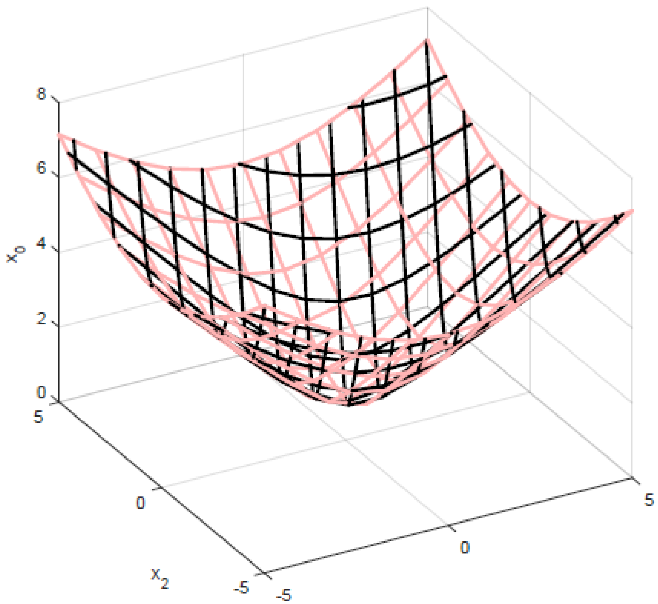


application of weighting metric *M*

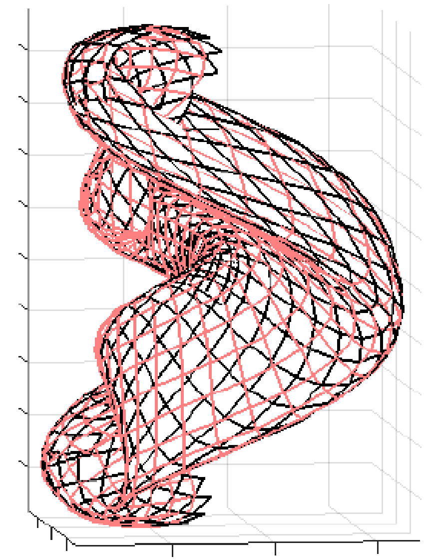
Can apply the same coordinate system transformation trick in curved spaces!

Metric Learning in non-Euclidean Spaces

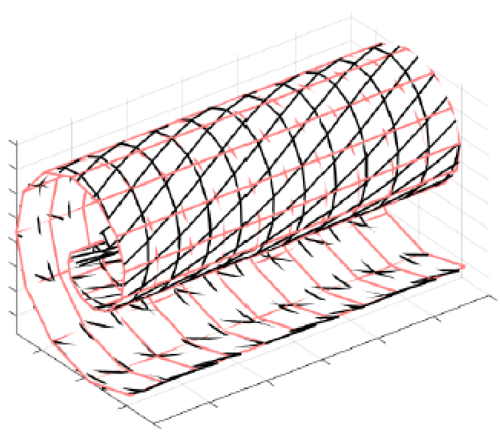
Hyperboloid space

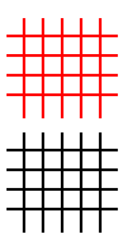


Kleinbottle surface



Swisroll





original coord. system

transformed coord. system

Learn Optimal Weighting

Want:

Distance metric: $\rho_M(\vec{x}_1, \vec{x}_2)$

Distances get computed wrt curved coord. system

such that: data samples from **same class** yield **small values**

data samples from different class yield large values

One way to solve it mathematically:

Create **two** sets: Similar $\vec{x}_j) \mid y_i = y_j\}$

HOW?

Dissim $\vec{x}_j) \mid y_i \neq y_j\}$

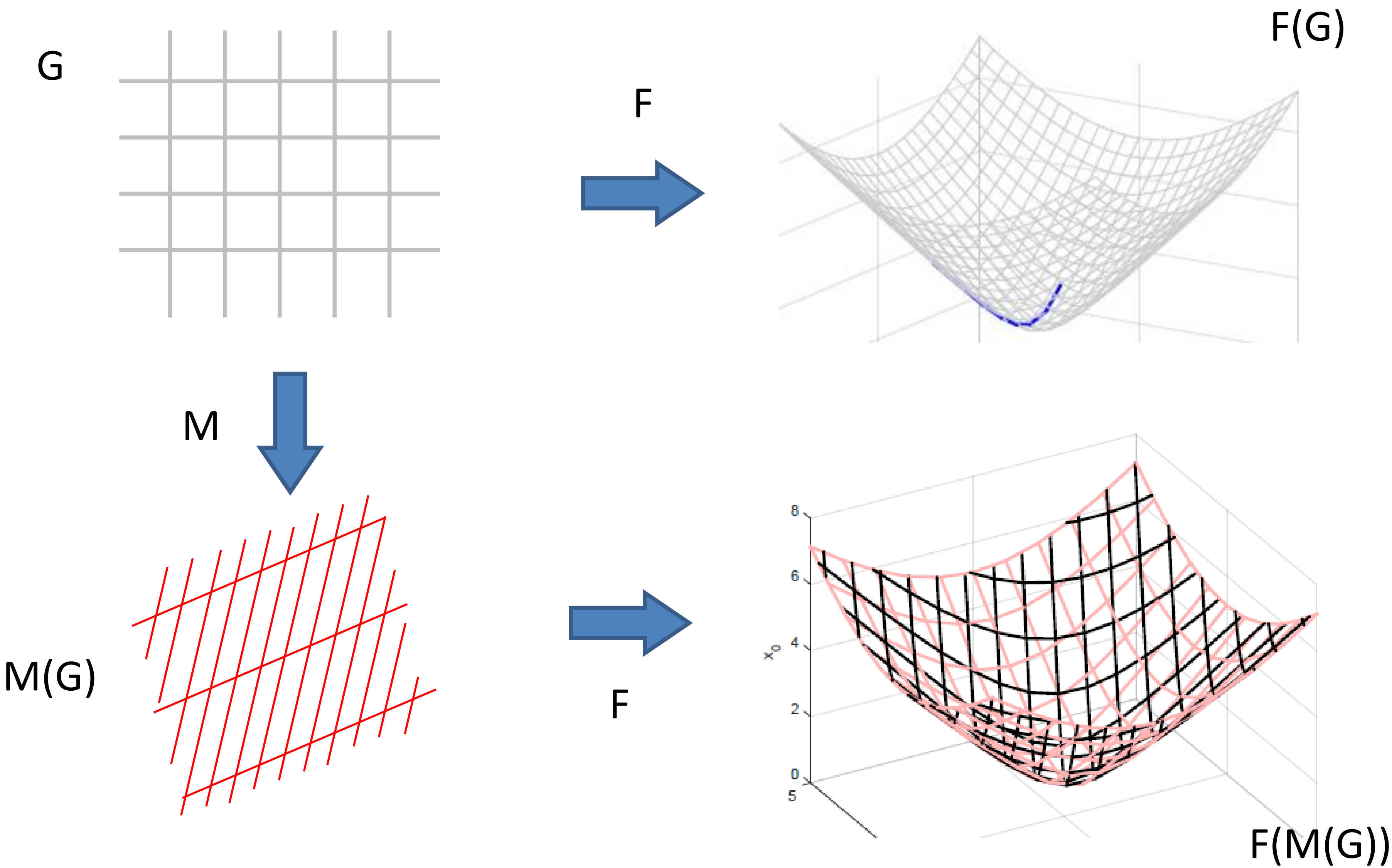
i, *j* = 1,…, *n*

Define a cost function:

$$\Psi(M) := \lambda \sum_{(\vec{x}_i, \vec{x}_j) \in S} \rho_M(\vec{x}_i, \vec{x}_j) \; - \; (1 - \lambda) \sum_{(\vec{x}_i, \vec{x}_j) \in D} \rho_M(\vec{x}_i, \vec{x}_j)$$

Minimize Ψ w.r.t. M!

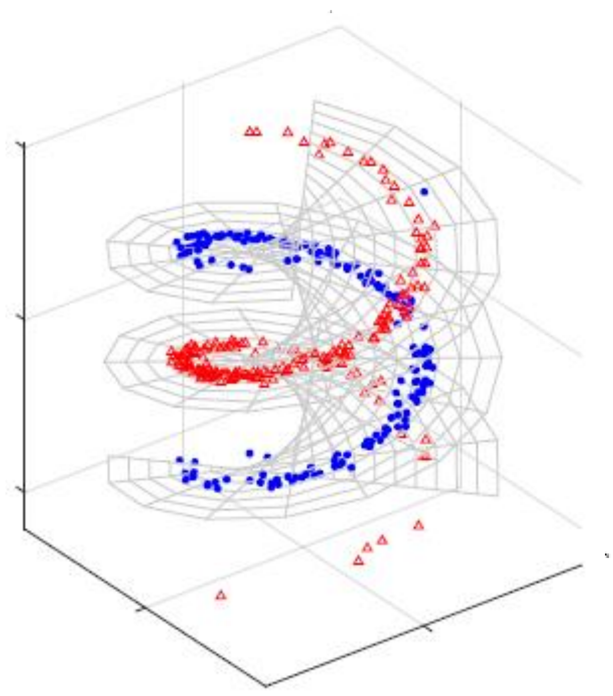
Transforming coordinates in non-Euclidean space



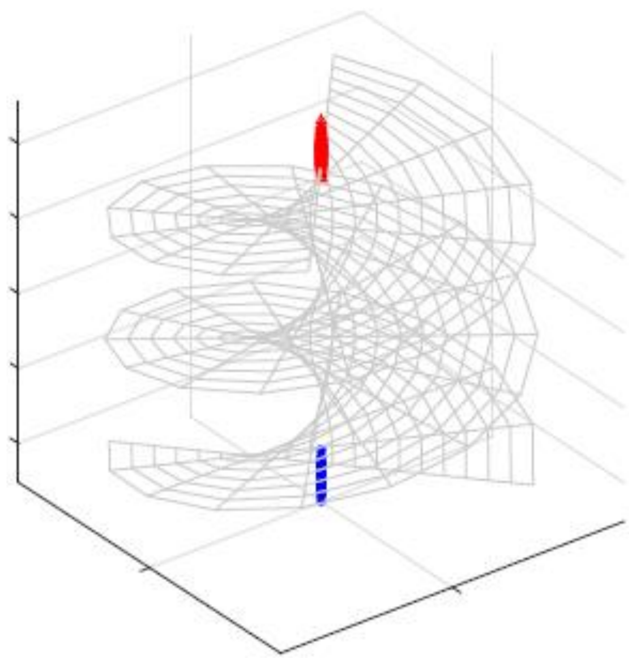
non-Euclidean spaces + Metric Learning

Using the **correct representation** followed by simple transformations and greatly **simplify the problem** difficulty!

[Aalto and Verma, 2019]



cannot separate the classes easily



Can easily separate the classes

Empirical performance

[Aalto and Verma, 2019]

Error of nearest neighbor classifier

Dataset	Euclidean	Euclidean+ Metric learn	Hyperbolic	Hyperbolic+ Metric Learn*
football	0.41 ± 0.09	0.40 ± 0.09	0.29 ± 0.09	**0.25 ± 0.10**
polbooks	0.24 ± 0.05	0.31 ± 0.12	0.25 ± 0.06	**0.23 ± 0.06**
adjnoun	0.58 ± 0.06	0.56 ± 0.07	0.55 ± 0.09	**0.49 ± 0.05**

Datasets:

- **football**: a network of American football teams, edges represent games, 12 categories for each division
- **polbooks**: books on US politics, edges represent co-purchase, 3 categories: ‘liberal’, ‘conservative’, ‘neutral’
- **adjnoun**: a network of words in Dicken’s novel, edges represent adjacent words, categories: ‘nouns’ and ‘adjectives’

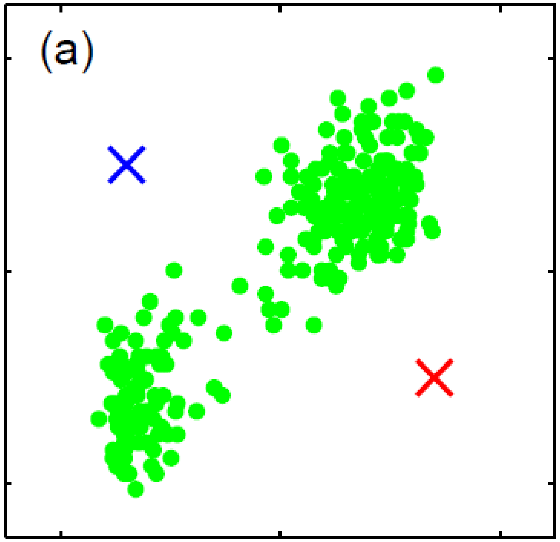
* Results comparable to better than state of the art reported results

Clustering in Euclidean spaces

Recall *k*-means clustering (aka Lloyd's method):

- Initialize *k* centers randomly

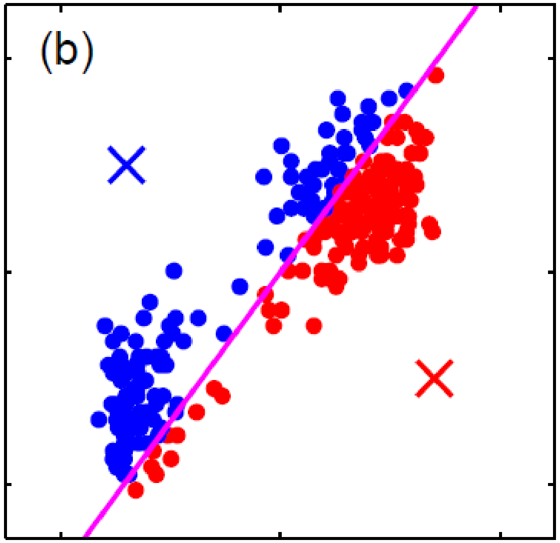
- Repeat until convergence
  - Partition the data wrt *k* centers
  - Recompute the centers for each partition by taking the mean



Clustering in Euclidean spaces

Recall *k*-means clustering (aka Lloyd's method):

- Initialize *k* centers randomly
- Repeat until convergence
  - Partition the data wrt *k* centers
  - Recompute the centers for each partition by taking the mean

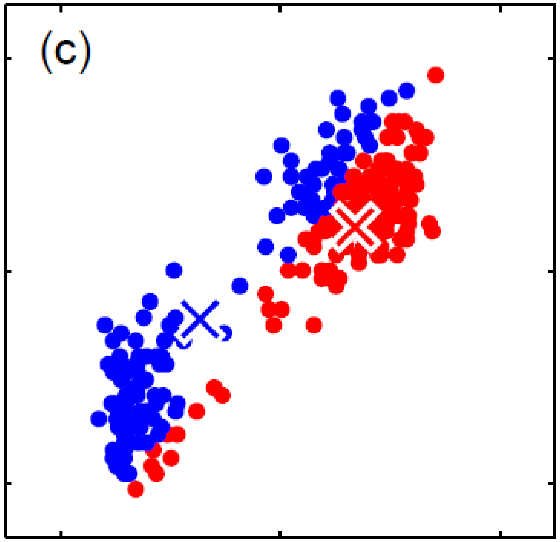


Clustering in Euclidean spaces

Recall *k*-means clustering (aka Lloyd's method):

- Initialize *k* centers randomly

- Repeat until convergence
  - Partition the data wrt *k* centers
  - Recompute the centers for each partition by taking the mean

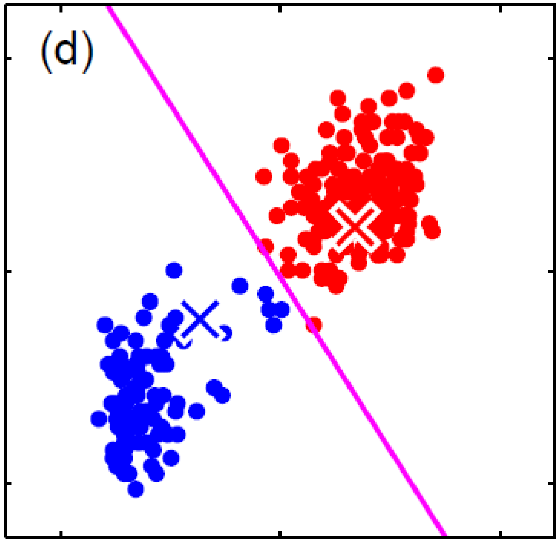


Clustering in Euclidean spaces

Recall *k*-means clustering (aka Lloyd's method):

- Initialize *k* centers randomly

- Repeat until convergence
  - Partition the data wrt *k* centers
  - Recompute the centers for each partition by taking the mean

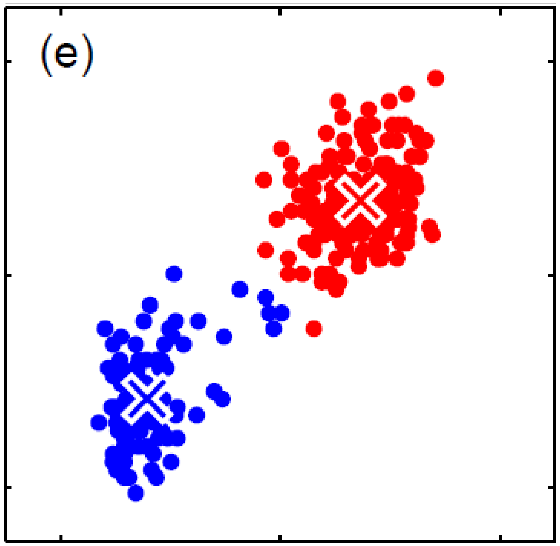


Clustering in Euclidean spaces

Recall *k*-means clustering (aka Lloyd's method):

- Initialize *k* centers randomly

- Repeat until convergence
  - Partition the data wrt *k* centers
  - Recompute the centers for each partition by taking the mean



Clustering in non-Euclidean spaces

Recall *k*-means clustering (aka Lloyd's method):

- Initialize *k* centers randomly
- Repeat until convergence
  - Partition the data wrt *k* centers
  - Recompute the centers for each partition by taking the mean

mean = $(1/n) \Sigma x_i$

vector addition is not possible!

now what?

Clustering in non-Euclidean spaces

Need to group datapoints, ***without*** computing the mean (or barycenter)

Observation:

- k-means minimizes the following objective function

$$\sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu_j\|^2$$

partitions $C_1, \ldots, C_j$

means $\mu_1, \ldots, \mu_j$

$$= \frac{1}{2|C_j|} \sum_{i,i' \in C_j} \|x_i - x_{i'}\|^2$$

$$\rho^2(x_i, x_{i'})$$

Clustering in non-Euclidean spaces

k-means clustering in non-Euclidean spaces:

- Randomly partition the data in *k* groups
- Repeat until no more improvement can be made
  - For each datapoint x_i and each partition C_j compute the k-means cost when x_i is assigned to cluster C_j

$$\left(\text{cost} = \sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{i,i' \in C_j} \rho^2(x_i, x_{i'}) \right)$$

Results show about 5% improvement in clustering quality on 20 newsgroup dataset

[Aalto and Verma, 2019]

Future directions

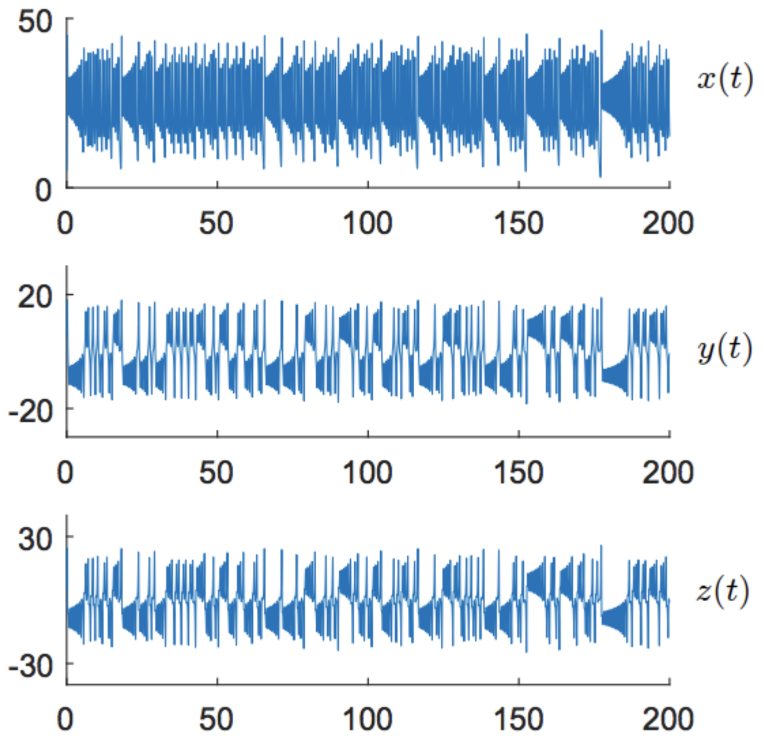
- Extend other ML algorithms

- Find effective representations of other interesting structured data

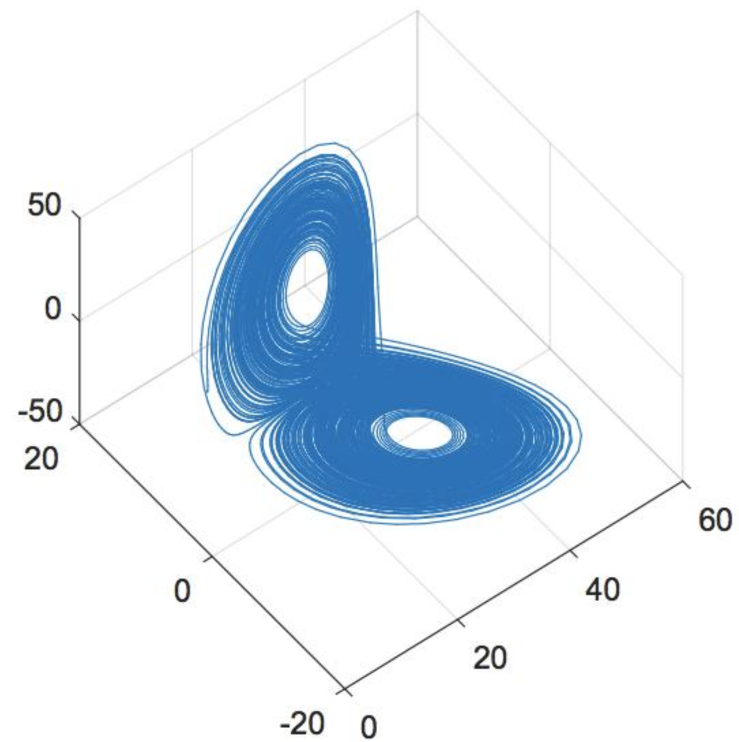
  time series data!

Future directions: time series

Time series data has many interesting patterns such as ***seasonality***



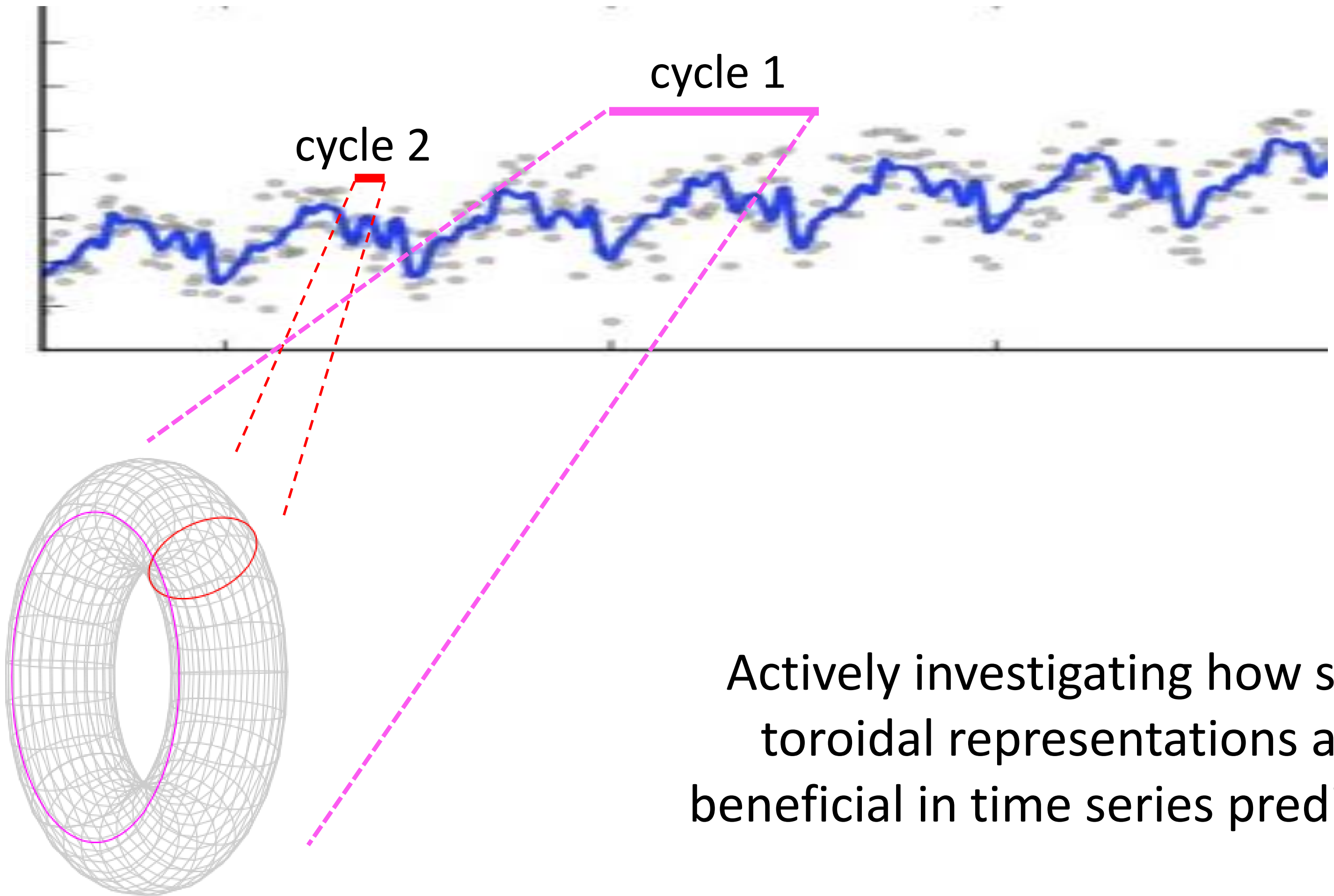
time series considered *separately*



time series considered ***jointly***

Future directions: time series

Seasonal patterns can be well represented in cyclical spaces



Actively investigating how such toroidal representations are beneficial in time series prediction

Questions/Discussion

Thank You!

for patiently listening! ☺

References

M. Aalto and N. Verma. Metric Learning on non-Euclidean spaces. In review. SIAM Data Science, 2019.

G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, A. On the local behavior of spaces of natural images. IJCV, 2008.

K. Weinberger and L. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification, JMLR 2008.