Transport Mode Recognition

Frédéric Chazal[1], Mathieu Feraille[2], Marc Glisse[1],
Kevin Jaunatre[2], Jisu Kim*[1], Guillaume Sabiron[2]

1 Inria DataShape

2 IFP Energies nouvelles

2020-12-01

Data and Problem

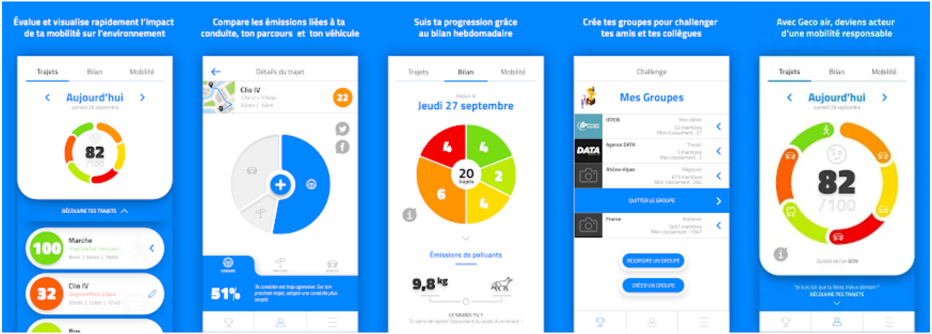
Featurization and Model: existing approach

Featurization and Model: new approach

Test results and Conclusion

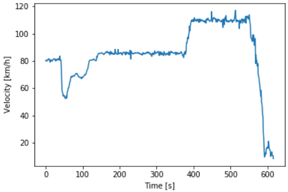
The application Geco air needs to classify trips according to their transport modes.

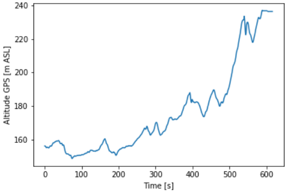
- Geco air is an application monitoring users' travels and advising to reduce pollutions.
- To do this, Geco air need to take users' trips as inputs and classify according to their transport modes (car, train, bus, walking, etc).



Data: GPS and accelerometer data of trips.

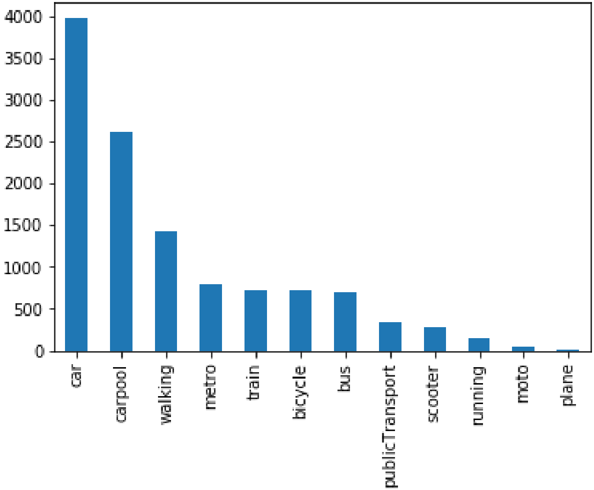
- ~70,000,000 trips from GPS and accelerometer data
  - several time series features: velocity, latitude, longitude, elevation
  - ~11500 trips selected for this problem



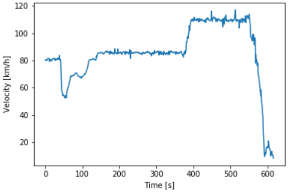


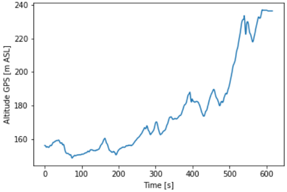
Data: GPS and accelerometer data of trips.

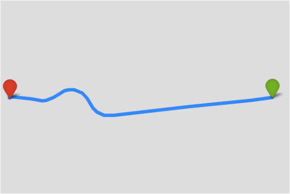
- Manually labeled by users as : car, carpool, bus, train, bicycle, moto, scooter, metro, running, walking, plane, public transport
- Some trips are multi-mode, i.e. mixture of several transportation modes



Example of well-defined trips

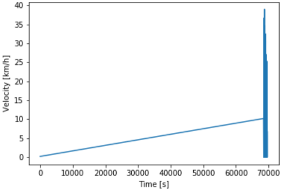


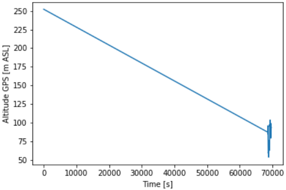


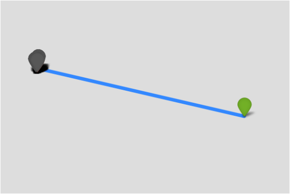


Example of less well-defined trips

- Less well-defined trips are due to: multi-modality, underground mobility, etc.







We classify transport modes.

- Goal: when a trip is received from Geco Air without transport mode, provide the transport mode with high reliability
- Classify 11500 trips to 9 classes (plane, car/carpool, bus, train, walking, bicycle/scooter, metro, moto, running)
- If the trip is difficult to classify, it's better to say that the trips is "unknown" rather than providing wrong transport mode
- Confusing between car and walking is more critical than confusing between car and bus

We use weighted accuracy tuned for our problem for the performance indicators.

- Unknown probability = Proportion of trips whose classification probability is less than 0.6
- Accuracy = Accuracy for the "known" trips, i.e. trips whose classification probability is more than 0.6.
- Score $= 0.85 \times$ (Accuracy) $+ 0.15 \times (1 -$ Unknown probability)
- Weighted Accuracy = Accuracy for known trips, with loss function below:
-

	car	bus	train	walking	bicycle	metro	moto	running
car	1	0.9	0.3	0	0	0	1	0
bus	0.9	1	0.4	0	0	0.2	0.9	0
train	0.3	0.4	1	0	0	0.4	0.3	0
walking	0	0	0	1	0.5	0	0	0.7
bicycle	0	0	0	0.5	1	0	0	0.5
metro	0	0.2	0.4	0	0	1	0	0
moto	1	0.9	0.3	0	0	0	1	0
running	0	0	0	0.7	0.5	0	0	1

- Weighted Score $= 0.85 \times$ (Weighted Accuracy) $+ 0.15 \times (1 -$ Unknown probability)

Data and Problem

Featurization and Model: existing approach

Featurization and Model: new approach

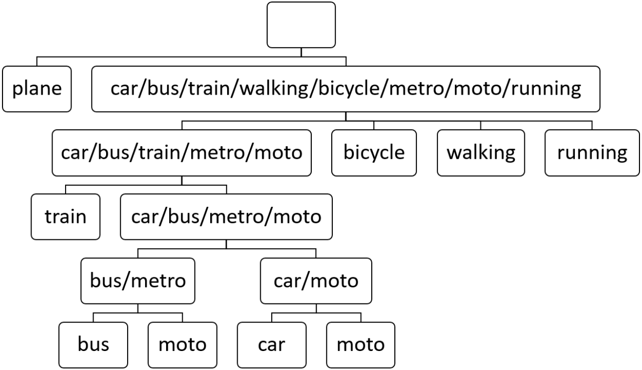
Test results and Conclusion

Several features are computed from raw data.

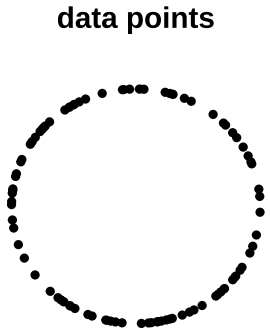
- From Arthur Mouchot's internship
- From raw data, several features are computed.
  - acceleration, number of stops, time between stops, velocity$\times$acceleration, etc.
  - for time series features, several additional features are computed: mean, median, 95% quantile, fourier transform, etc.

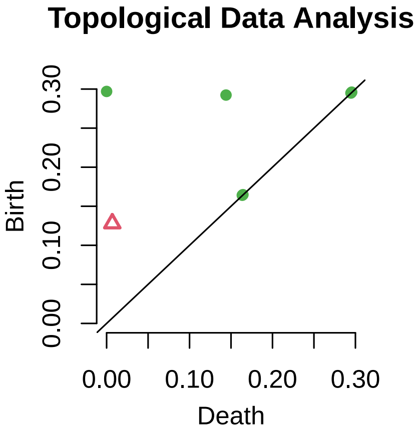
We use 5 units model.

- From Arthur Mouchot's internship
- We use random forest for each unit classifier.



Topological Data Analysis quantifies and extracts topological information from data.





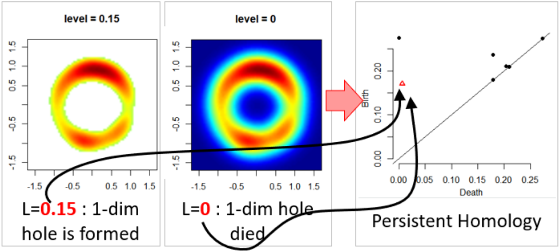
We additionally consider two featurization using Topological Data Analysis.

- Since we have several 1-dimensional time series features, we can apply topological data analysis.
- We consider two featurization from velocity, acceleration, velocity$\times$acceleration:
  - Sub/super-level filtration + Topological Data Analysis
  - Time-delayed embedding + Topological Data Analysis

Proposed Featurization for gps velocity: Sub/super-level filtration + Topological Data Analysis

- Input: time series data $x = \{x_0, \cdots, x_N\} \subset \mathbb{R}$, Output: vector $\lambda^k \in \mathbb{R}^k$.

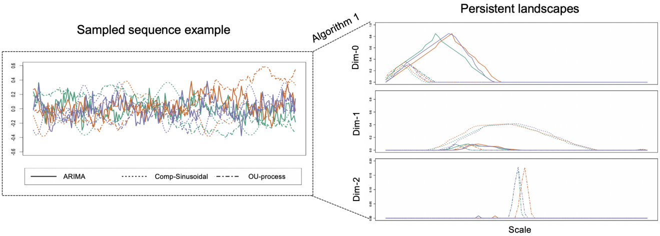
1. Construct the sub-level filtration x_{sub} and compute the persistence diagram $Dgm(x_{sub})$.
2. From $Dgm(x_{sub})$, compute the landscape $\lambda_{sub} : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$.
3. Construct the super-level filtration x_{super} and compute the persistence diagram $Dgm(x_{super})$.
4. From $Dgm(x_{super})$, compute the landscape $\lambda_{super} : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$.
5. Vectorize λ_{sub} and λ_{super} to get $\lambda^K \in \mathbb{R}^K$.
6. Perform PCA on λ^K and get $\lambda^k \in \mathbb{R}^k$.



Proposed featurization for gps velocity: Time-delayed embedding + Topological Data Analysis

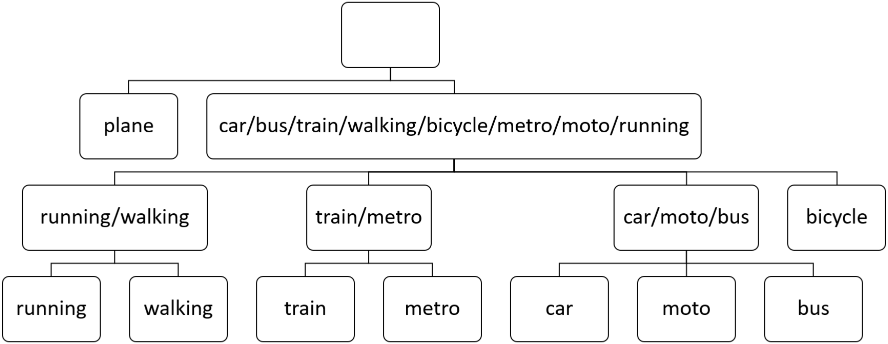
- Input: time series data $x = \{x_0, \cdots, x_N\} \subset \mathbb{R}$, Output: k-dimensional vector $\lambda^k \in \mathbb{R}^k$.

1. Construct the point cloud $X \subset \mathbb{R}^m$ using the time-delayed embedding with parameters m, τ.
2. Perform PCA on X and obtain $X^\ell \subset \mathbb{R}^l$.
3. Construct the Rips filtration R_{X^l} and compute the persistence diagram $Dgm(X^l)$.
4. From $Dgm(X^l)$, compute the landscape $\lambda : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$.
5. Vectorize λ to get $\lambda^K \in \mathbb{R}^K$.
6. Perform PCA on λ^K and get $\lambda^k \in \mathbb{R}^k$.



We use 4 units model.

- We tested several models and choose this 4 units model.
- We use random forest for each unit classifier.



Test result for 4 units model.

- We use K-fold to test model.

	Unknown probability	Accuracy	Weighted Accuracy	Score	Weighted Score
5 units (Random Forest)	0.430	0.883	0.920	0.836	0.868
4 units + TDA (Random Forest)	0.520	0.918	0.944	0.853	0.874

Mean confusion matrix for 4 units model

	car	bus	train	walking	bicycle	metro	moto	running	**recall**
car	160.9	2.2	2.0	1.0	2.3				**0.95**
bus	2.9	2.2		0.6	0.7				**0.36**
train	0.5		11.2		0.4				**0.93**
walking	1.1		0.3		0.5			0.5	**0.93**
bicycle	0.4			1.0	40.0			0.5	**0.96**
metro	0.2	0.4	0.2	1.6	0.8	0.5			**0.16**
moto									
running				1.3	0.6			2.9	**0.60**
precision	**0.97**	**0.47**	**0.83**	**0.85**	**0.89**	**0.4**		**0.78**	**0.92**

Conclusion

- The classification problem comes from improving Geco air application.
- The classification problem has several difficulties: ill-defined trips, multi-modal trips, imbalanced labels, etc
- We used weighted score tuned for our prolem.
- By utilizing topological data analysis, we could have improved the classification result.
- The quality of the classification result greatly depends on the labels.

Future work

- For this project: further applying topological data analysis, applying other machine learning frameworks, taking a closer look at misclassified data, etc.
- Inria Datashape - Ifpen will also work on another project: predicting permiability of rock

Thank you!

Background

Proposed Featurization for gps velocity: Sub/super-level filtration + Topological Data Analysis

Proposed featurization for gps velocity: Time-delayed embedding + Topological Data Analysis

Model Selection

Test results for model

Future Plans

Number of holes is used to summarize Topological features.

- Geometrical objects :
  - A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
  - 가, 字, あ
- Number of holes of different dimensions is considered.

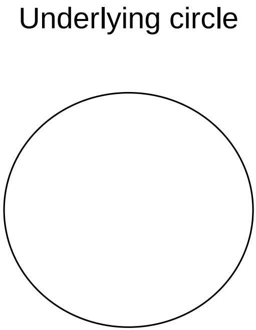
1. $\beta_0 = \#$ of connected components 
2. $\beta_1 = \#$ of loops (holes inside 1-dim sphere) 
3. $\beta_2 = \#$ of voids (holes inside 2-dim sphere) : if *dim* ≥ 3 

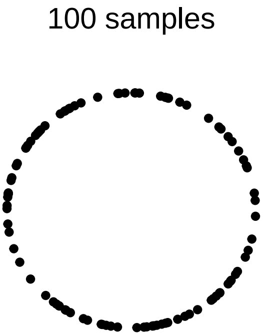
Example : Objects are classified by homologies.

1. β_0 =# of connected components
2. β_1 =# of loops

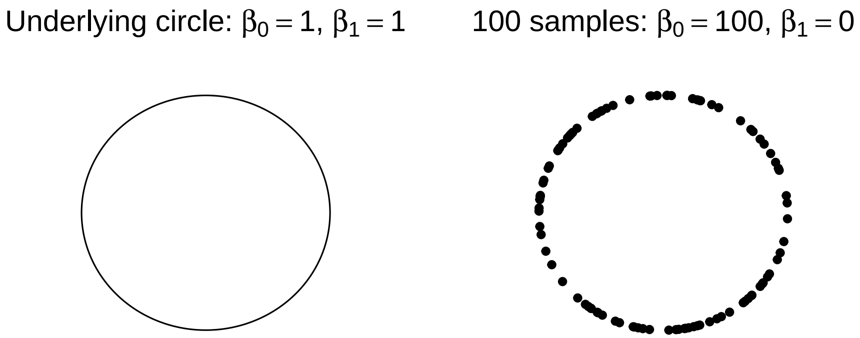
$\beta_0 \setminus \beta_1$	0	1	2
1	C, G, I, J, L, M, N, S, U, V, W, Z, E, F, T, Y, H, K, X	A, R, D, O, P, Q	B, あ
2	가, 字		

When analyzing data, we prefer robust features where features of the underlying manifold can be inferred from features of finite samples.

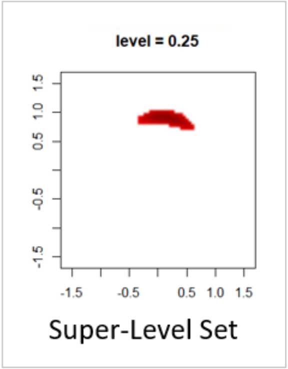




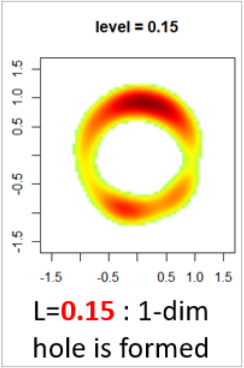
Homology of finite sample is different from homology of underlying manifold, hence it cannot be directly used for the inference.

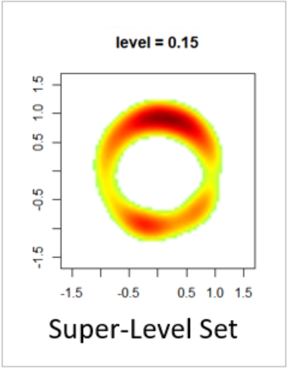


Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.

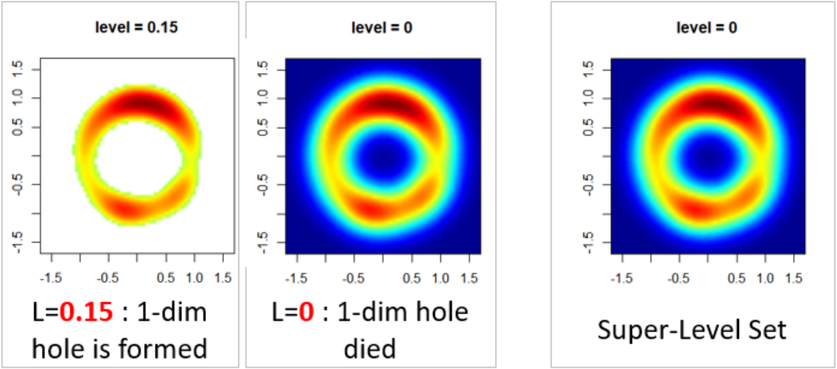


Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.

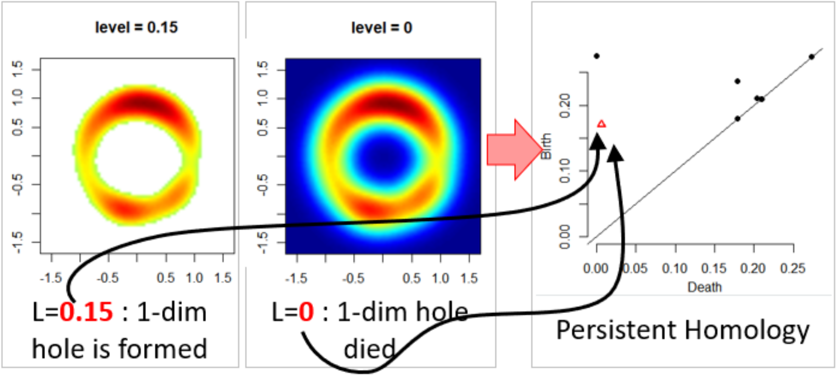




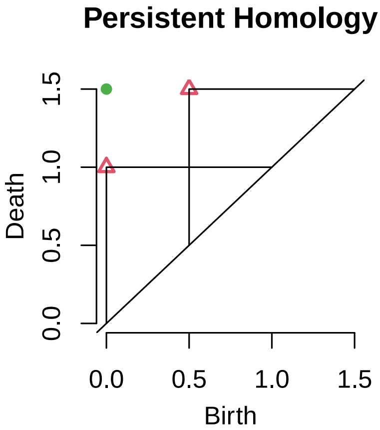
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.

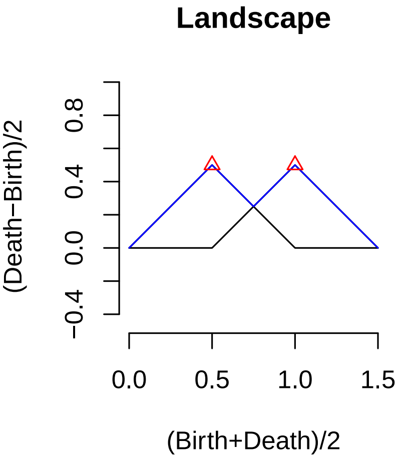


Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



Landscape is a functional summary of the persistent homology.





Background

Proposed Featurization for gps velocity: Sub/super-level filtration + Topological Data Analysis

Proposed featurization for gps velocity: Time-delayed embedding + Topological Data Analysis

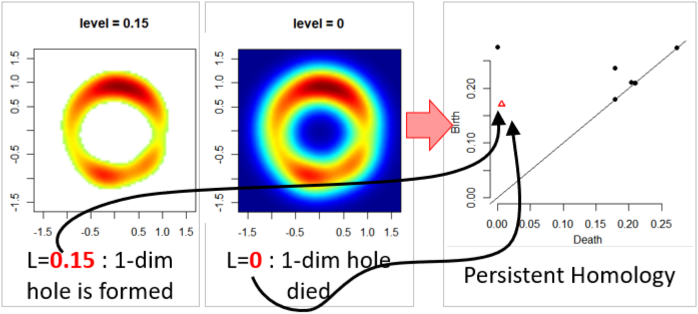
Model Selection

Test results for model

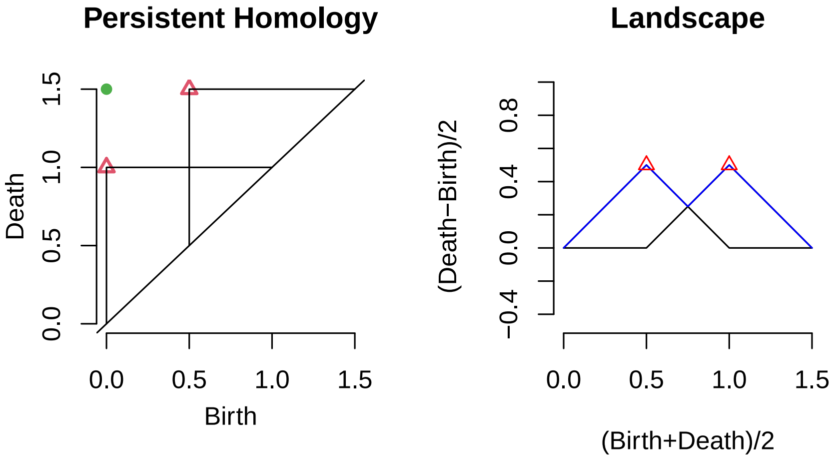
Future Plans

1. Construct the sub-level filtration x_{sub} and compute the persistence diagram $Dgm(x_{sub})$.

- For $f : \mathbb{R}^d \to \mathbb{R}$ and $r > 0$, the sub-level set is
$$f^{-1}(-\infty, r] = \{x \in \mathbb{R}^d : f(x) \leq r\}.$$
- For the time-series data x, we constrcut a function $f_x : \mathbb{R} \to \mathbb{R}$ by a piecewise linear function having (i, x_i) as vertices. Then we make the sub-level filtration as
$$\{x_{sub}(r)\}_{r>0} = \{f_x^{-1}(-\infty, r]\}_{r>0},$$
and compute its persistence diagram $Dgm(x_{sub})$.



2. From $Dgm(x_{sub})$, compute the landscape $\lambda_{sub} : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$.



3. Construct the super-level filtration x_{super} and compute the persistence diagram $Dgm(x_{super})$.

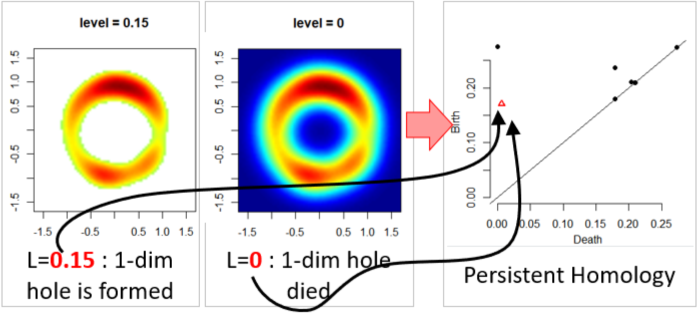
- For $f : \mathbb{R}^d \to \mathbb{R}$ and $r > 0$, the super-level set is

$$f^{-1}[r, \infty) = \left\{ x \in \mathbb{R}^d : f(x) \geq r \right\}.$$

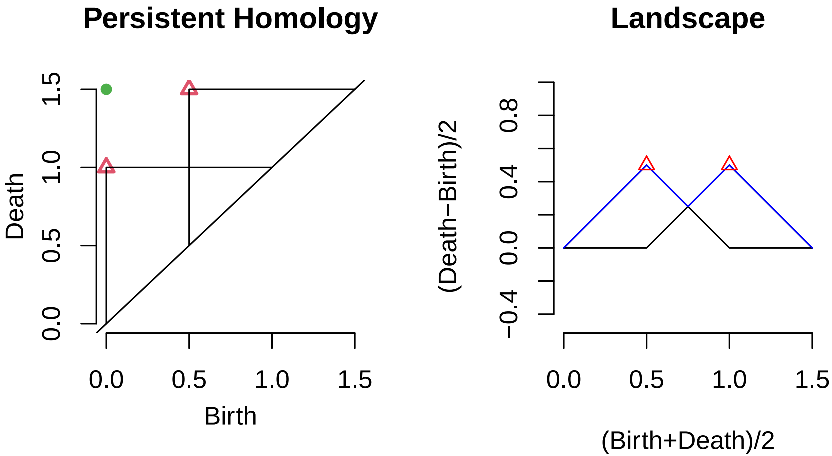
- For the time-series data x, we constrcut a function $f_x : \mathbb{R} \to \mathbb{R}$ by a piecewise linear function having (i, x_i) as vertices. Then we make the sub-level filtration as

$$\left\{ x_{super}(r) \right\}_{r>0} = \left\{ f_x^{-1}[r, \infty) \right\}_{r>0},$$

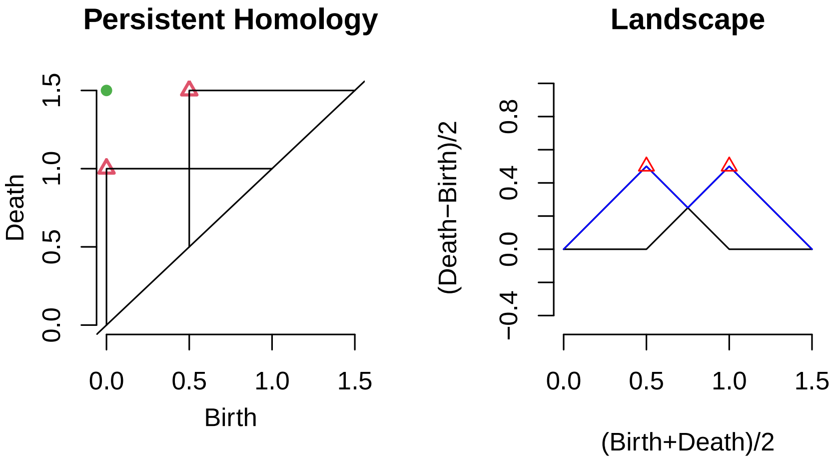
and compute its persistence diagram $Dgm(x_{super})$.



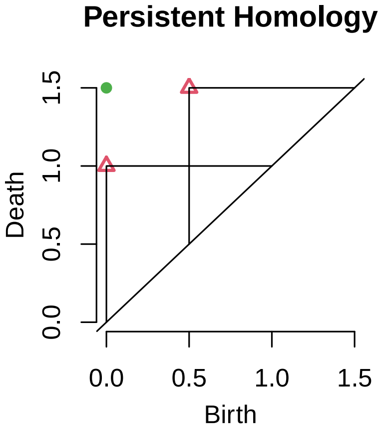
4. From $Dgm(x_{super})$, compute the landscape $\lambda_{super} : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$.

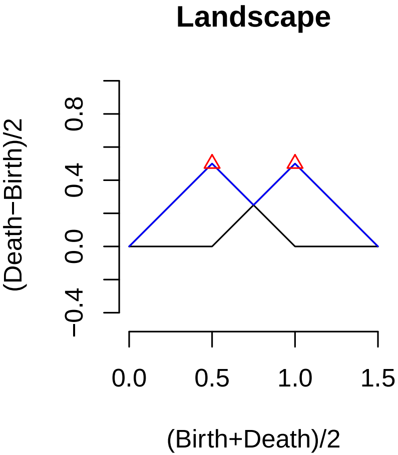


5. Vectorize λ_{sub} and λ_{super} to get $\lambda^K \in \mathbb{R}^K$.



6. Perform PCA on λ^K and get $\lambda^k \in \mathbb{R}^k$.

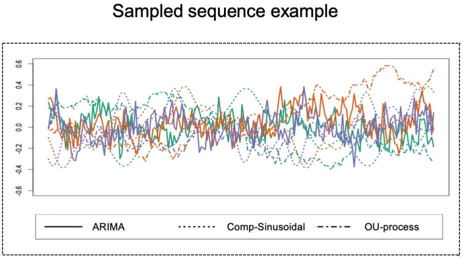


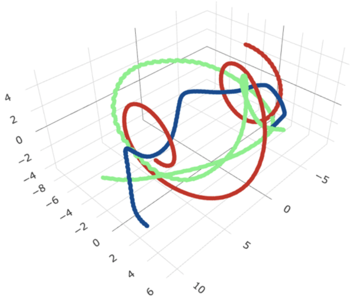


1. Construct the point cloud $X \in \mathbb{R}^m$ using the time-delayed embedding with parameters m, τ.

Let f be the time series function. Then, let the sliding window mapping $SW_{m,\tau}f : \mathbb{R} \to \mathbb{R}^m$ be

$$SW_{m,\tau}f(t) := [f(t-(m-1)\tau), \dots, f(t-\tau), f(t)]^\top.$$



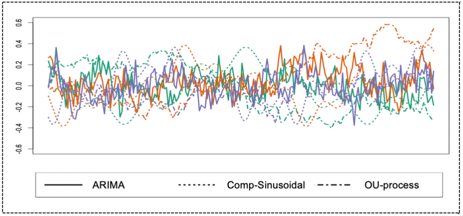


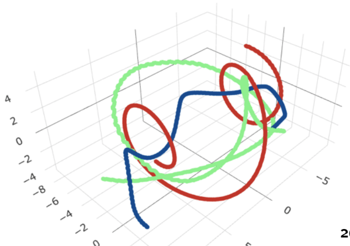
1. Construct the point cloud $X \subset \mathbb{R}^m$ using the time-delayed embedding with parameters m, τ.

Let $x_0, x_1, \ldots, x_N$ be a sequence of equi-interval samples from the time series function f, with $x_0 = f(0)$ and $x_N = f(T)$. Then, we construct the trajectory matrix X as

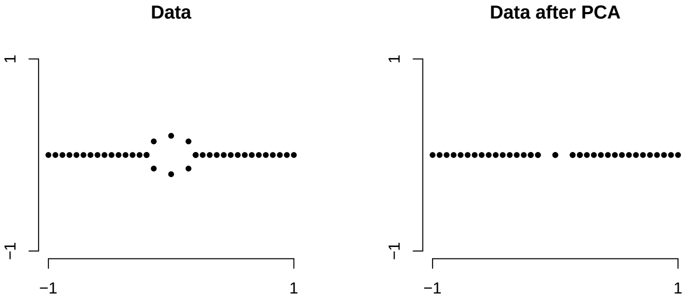
$$X = \begin{bmatrix} SW_{m,\tau} f((m-1)\tau) \\ SW_{m,\tau} f(1+(m-1)\tau) \\ \vdots \\ SW_{m,\tau} f(T) \end{bmatrix} = \begin{bmatrix} x_0 & x_\tau & \cdots & x_{(m-1)\tau} \\ x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-(m-1)\tau} & x_{N-(m-2)\tau} & \cdots & x_N \end{bmatrix}.$$

Sampled sequence example





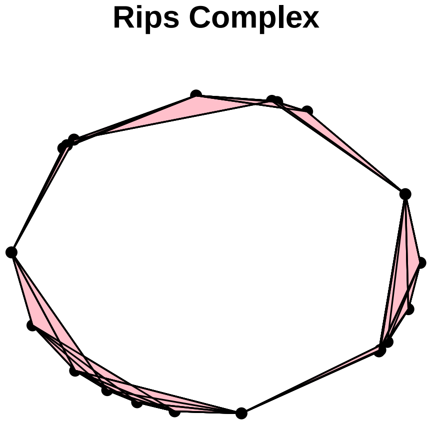
2. Perform PCA on X and obtain $X^\ell \subset \mathbb{R}^l$.



3. Construct the Rips filtration R_{X^l} and compute the persistence diagram $Dgm(X^l)$.

- For $\mathcal{X} \subset \mathbb{R}^d$ and $r > 0$, the Rips complex $\mathrm{Rips}(\mathcal{X}, r)$ is defined as

$$\mathrm{Rips}(\mathcal{X}, r) = \{\{x_1, \ldots, x_k\} \subset \mathcal{X} : d(x_i, x_j) < 2r, \text{ for all } 1 \leq i, j \leq k\}.$$

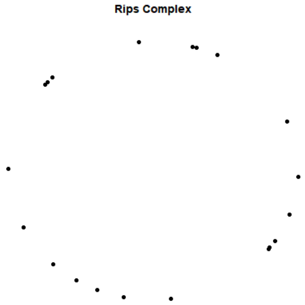


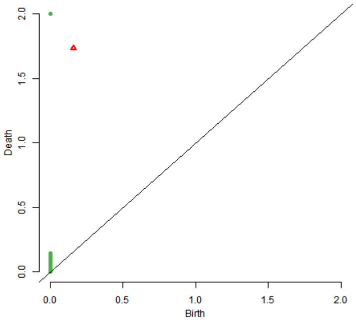
3. Construct the Rips filtration R_{X^l} and compute the persistence diagram $Dgm(X^l)$.

- For the dataset X^l, we make the Rips filtration as

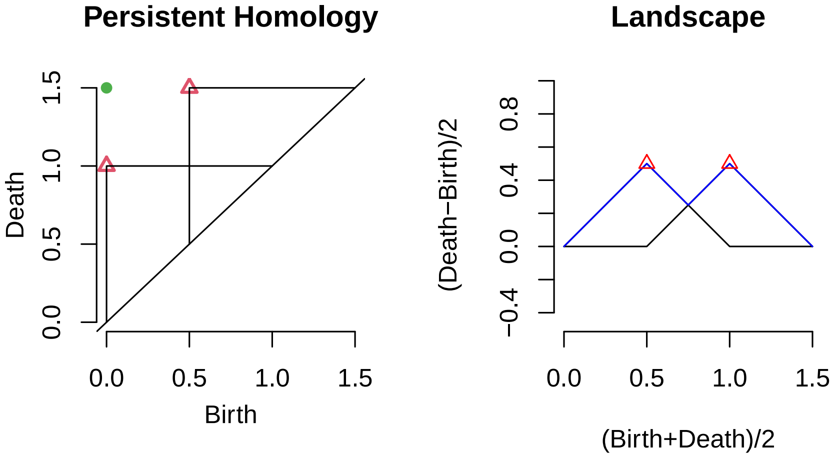
$$\{R_{X^l}(r)\}_{r>0} = \{\mathrm{Rips}(X^l, r)\}_{r>0},$$

and compute its persistence diagram $Dgm(X^l)$.

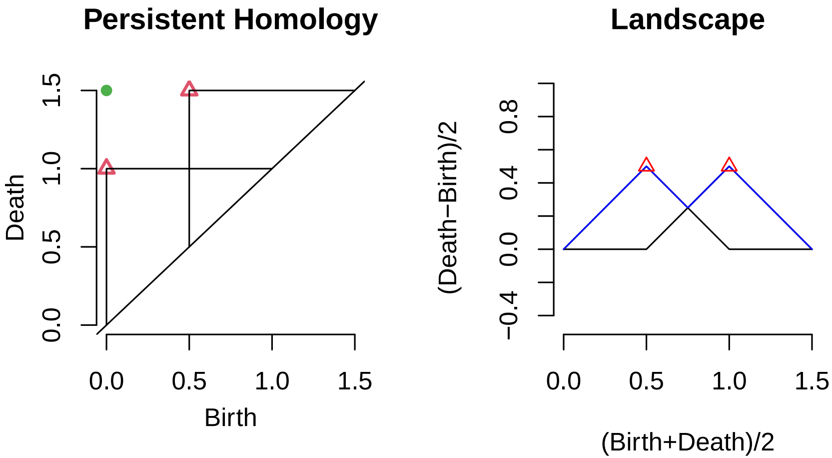




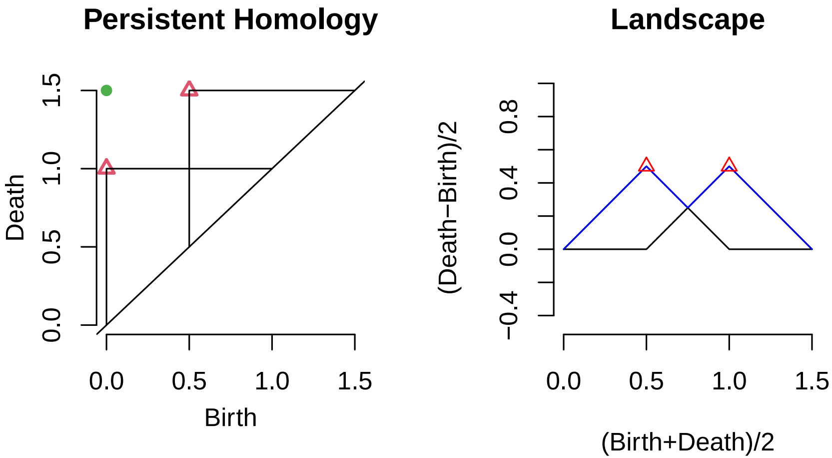
4. From $Dgm(X^l)$, compute the landscape $\lambda : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$.



5. Vectorize λ to get $\lambda^K \in \mathbb{R}^K$.



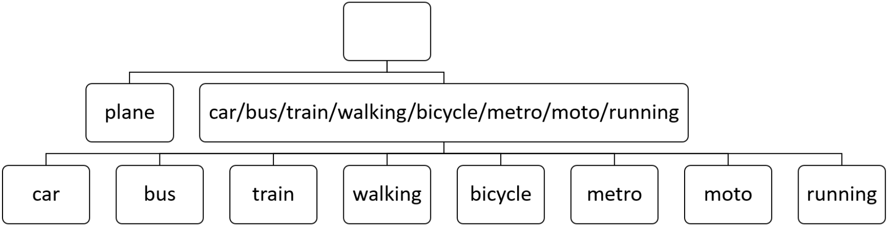
6. Perform PCA on λ^K and get $\lambda^k \in \mathbb{R}^k$.



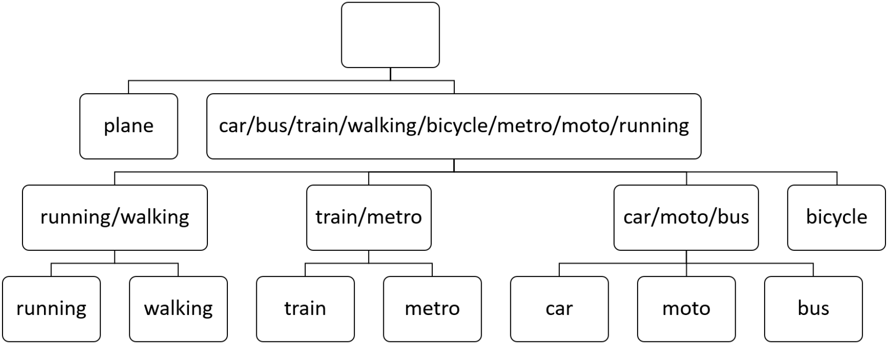
We compare 3 different models.

- We separate out plane first, and then apply different models on the rest.
- We compare 3 different models: 1 unit model, 4 units model, 5 units model.
- For each unit classifier, we tried different algorithms (random forest, Adaboost, Xgboost, SVM, logistic regression, naive Bayes, Knn) and choose the one with the best accuracy.

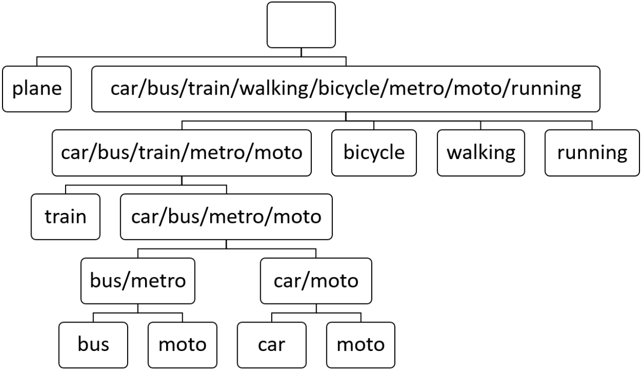
We compare 3 different models : 1 unit model



We compare 3 different models : 4 units model



We compare 3 different models : 5 units model



Background

Proposed Featurization for gps velocity: Sub/super-level filtration + Topological Data Analysis

Proposed featurization for gps velocity: Time-delayed embedding + Topological Data Analysis

Model Selection

Test results for model

Future Plans

We choose algorithm for each classifier based on test results.

- We use K-fold to test algorithms for classifiers.
- Conclusion:
  - bus/metro - carpool/car/moto : Logistic Regression
  - bus - metro : XgBoost
  - All the others: Random Forest

We choose 4 units model based on test results.

- We use K-fold to test models.
- Conclusion: use 4 units model.

	Unknown probability	Accuracy	Weighted Accuracy	Score	Weighted Score
1 unit (Random Forest)	0.382	0.905	0.936	0.862	0.888
4 units (Random Forest)	0.385	0.909	0.938	0.865	0.890
5 units (RF+ LR + XB)	0.289	0.882	0.920	0.856	0.888

TDA features lead to higher scores.

- We additionally compare 4 units with TDA vs 4 units without TDA.
- Conclusion: use TDA features

	Unknown probability	Accuracy	Weighted Accuracy	Score	Weighted Score
4 units, with TDA (Random Forest)	0.385	0.909	0.938	0.865	0.890
4 units, without TDA (Random Forest)	0.376	0.900	0.933	0.859	0.887

Background

Proposed Featurization for gps velocity: Sub/super-level filtration + Topological Data Analysis

Proposed featurization for gps velocity: Time-delayed embedding + Topological Data Analysis

Model Selection

Test results for model

Future Plans

Future Plans

There are several things to try, including:

- Further fine tuning Topological Data Analysis features
- Spliting a multi-modal trip to several unimodal trips
- Merge trips to generate more trips
- Taking a closer look at misclassified data
- Incorporating locations of public transport
- Applying other machine learning frameworks (e.g., deep learning)

Further fine tuning Topological Data Analysis features

- Topological Data Analysis requires several parameters to choose, in particular with time-delayed embedding.

Spliting a multi-modal trip to several unimodal trips

- Some trips are multi-mode, i.e. mixture of several transportation modes
- For this trip, detecting change of transportation mode and spliting it to several unimodal trips can help to improve classification.

Merge trips to generate more trips

- Merge several trips with the same transport mode to generate more trips.

Taking a closer look at misclassified data

- Taking a closer look at where the classifier fails (e.g., bus-car, metro-car, running-bicycle) will help designing features.

	car	bus	train	walking	bicycle	metro	moto	running	recall
car	160.9	2.2	2.0	1.0	2.3				0.95
bus	2.9	2.2		0.6	0.7				0.36
train	0.5		11.2		0.4				0.93
walking	1.1		0.3		0.5			0.5	0.93
bicycle	0.4			1.0	40.0			0.5	0.96
metro	0.2	0.4	0.2	1.6	0.8	0.5			0.16
moto									
running				1.3	0.6			2.9	0.60
precision	0.97	0.47	0.83	0.85	0.89	0.4		0.78	0.92

Applying other machine learning frameworks (e.g., deep learning)

- Instead of random forest, we can apply other machine learning frameworks, for example, deep learning.

Incorporating locations of public transport

- Train, bus, metro have fixed networks.
- If we can incorporate geographical locations of these networks, the classifier can be improved.