NeurIPS 2024

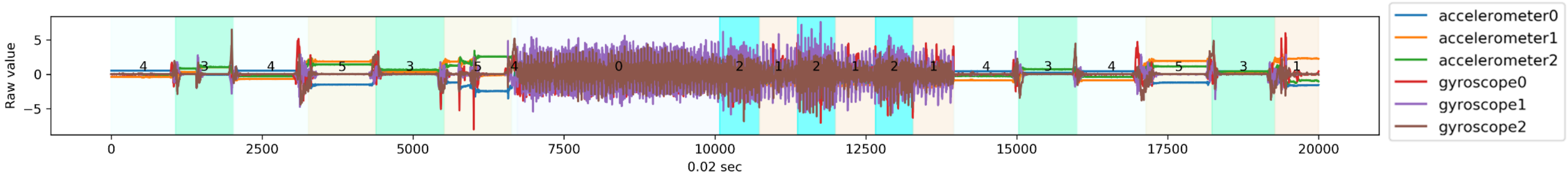
Exploiting Representation Curvature for Boundary Detection in Time Series

Yooju Shin[1] Jaehyun Park[1] Hwanjun Song[1] Susik Yoon[2] Jae-Gil Lee[1] Byung Suk Lee[3]

[1]KAIST [2]Korea University [3]University of Vermont

Time Series Boundary Detection

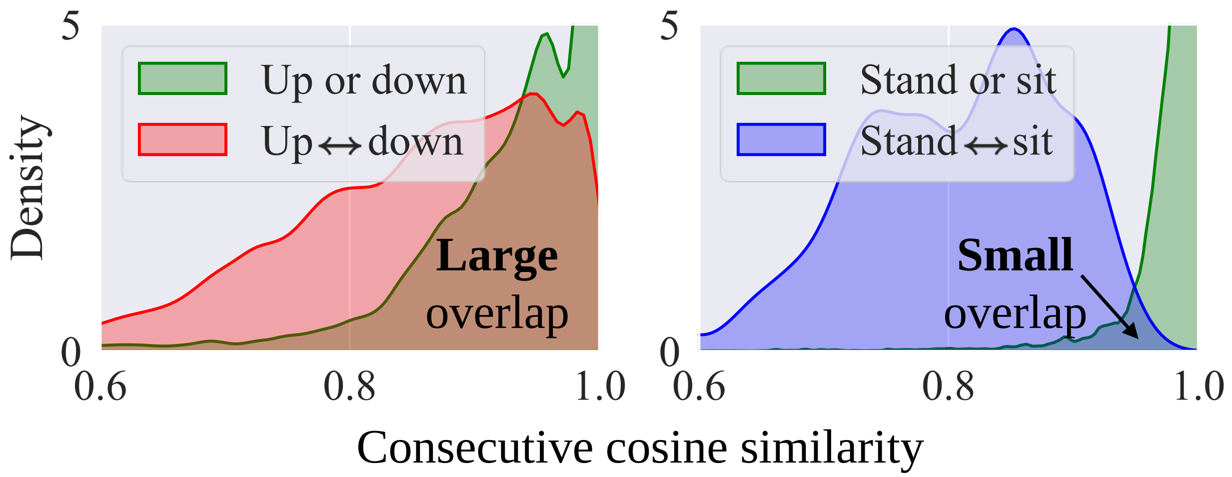
- A time series is a sequence of coherent data points
- Boundaries signify class transitions in time series data
  - The shade represents each action class such as running and walking
- Boundary detection helps accurate prediction and monitoring



An Example: Human Activity Recognition Dataset

Limitation in Boundary Detection

- However, traditional methods rely on consecutive distance
- **Subtle or gradual changes are often missed!**
- The figure shows the density of consecutive cosine similarity
  - The large overlap indicates the challenge in accurate detection

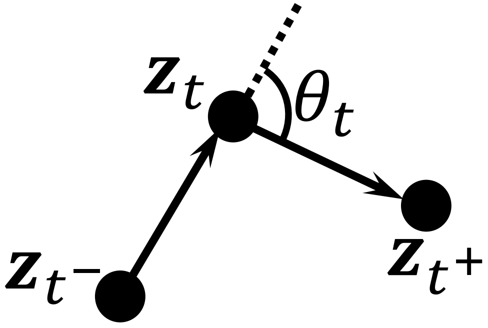


RECURVE: Curvature-based Method

- Curvature measures how sharply the trajectory changes direction
- Curvature is computed from three representation points
  - $\boldsymbol{z}_{t^-}$, $\boldsymbol{z}_t$, and $\boldsymbol{z}_{t^+}$ are the representation points where $t^- < t < t^+$
- Higher curvature occurs when the trajectory makes sharp turns
  - Turning angle is big and consecutive distances are small

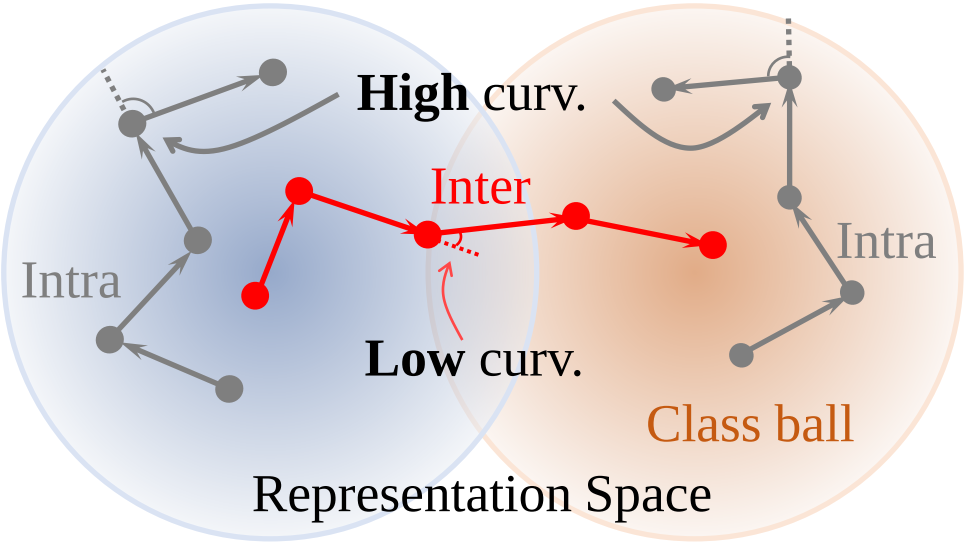
$$\theta_t = \arccos \frac{(\boldsymbol{z}_t - \boldsymbol{z}_{t^-}) \cdot (\boldsymbol{z}_{t^+} - \boldsymbol{z}_t)}{\|\boldsymbol{z}_t - \boldsymbol{z}_{t^-}\| \|\boldsymbol{z}_{t^+} - \boldsymbol{z}_t\|}$$

$$\kappa_t = \frac{\theta_t}{\|\boldsymbol{z}_t - \boldsymbol{z}_{t^-}\| + \|\boldsymbol{z}_{t^+} - \boldsymbol{z}_t\|}$$



Rationale behind Curvature

- Representation learning clusters class representations as a ball
- Due to confinement, intra-seg. points show high curvature
  - In contrast, inter-seg. points do show low curvature
- Despite of similar distances, curvature still detects changes



Evaluation of RECURVE

- RECURVE is compared to three recent works using four datasets
  - Two representation learning methods are used: TPC and TNC
- It detects the closest point to true change points

Methods	LOC ↓ (thresholding by best F1)				LOC ↓ (thresholding by mean segment length)			
	WISDM	HAPT	mHealth	50salads	WISDM	HAPT	mHealth	50salads
RuLSIF	420.9±18.54	108.2±0.188	780.0±8.580	184.4±1.463	429.5±9.968	156.0±0.092	802.6±30.18	189.2±1.120
KL-CPD	189.0±12.20	121.5±4.540	306.4±126.5	179.5±3.853	198.3±2.329	113.0±2.545	352.6±119.7	176.6±1.017
TS-CP^2	166.6±7.840	386.6±31.04	879.4±62.57	119.0±6.712	183.1±15.13	404.2±32.60	923.8±44.39	129.4±5.091
RECURVE+TPC	**114.7±56.07**	**33.25±1.290**	483.6±64.24	**79.29±10.52**	**178.4±36.05**	**34.28±0.727**	341.0±47.93	**93.76±7.475**
RECURVE+TNC	210.0±112.3	47.92±2.884	**224.0±211.2**	175.0±26.38	219.8±102.2	50.71±1.589	**239.6±212.4**	178.8±20.87

Conclusion

- We propose **RECURVE**, exploiting curvature of repr. trajectory
- RECURVE is simple and effective, used with any learned representation
- RECURVE enhances accuracy of CPD by up to **12.7% without any label**

EOF