MANTRA : A Scalable Approach to Mining Temporally Anomalous Sub-trajectories

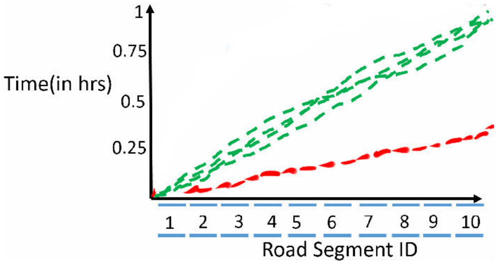
ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016)

Prithu Banerjee, Pranali Yawalkar, Sayan Ranu

17 June 2016

Introduction : Mining Temporally Anomalous Sub-Trajectories

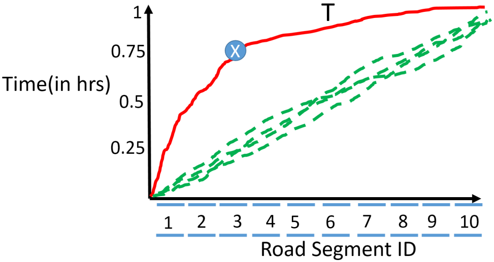
Temporally **Anomalous**



- Time taken to cover the trajectory deviates ***significantly*** from the remaining population
- Both ***over-speeding*** and ***under-speeding*** are anomalous

Introduction : Mining Temporally Anomalous Sub-Trajectories

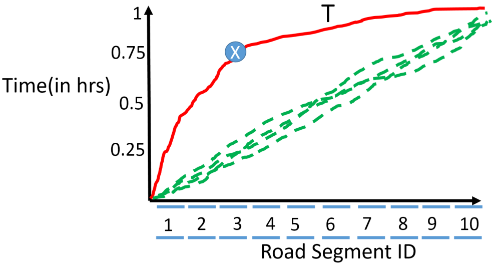
Why mine temporally anomalous *sub-trajectories*?



A non-anomalous trajectory may contain temporally anomalous sub-trajectories

Introduction : Mining Temporally Anomalous Sub-Trajectories

Why mine *maximal* anomalies ?



- No extra information provided by $T[1:2]$, $T[5:9]$ over $T[1:3]$ and $T[4:10]$
- Identifying ***longest stretches*** of anomalous driving

Applications

- **Real Time Vehicle Monitoring** :
  - Abundance of ***GPS data*** from smart devices ; MANTRA works ***directly*** on users' GPS data
  - Identifying ***anomalous drivers*** (over-speeding & under-speeding) in real time ; $<$ ***25 ms***
  - Robust against all year ***weather & traffic conditions*** ; useful in countries like **India**

Applications

- **Real Time Vehicle Monitoring :**
  - Abundance of ***GPS data*** from smart devices ; MANTRA works ***directly*** on users' GPS data
  - Identifying ***anomalous drivers*** (over-speeding & under-speeding) in real time ; $<$ ***25 ms***
  - Robust against all year ***weather & traffic conditions*** ; useful in countries like **India**
- **Identifying Bus Bunching**
  - A common phenomenon in countries like ***India***
  - Buses do not maintain the distance between the following ones ***uniformly***
  - ***Undesirable*** behaviour of ***under-speeding*** to pick up maximum passengers followed by ***over-speeding*** to maintain the distance from the following bus

Applications

- **Rating cab drivers** :
  - ***GPS trackers*** already installed in all cabs
  - Identifying ***how*** anomalous and ***where*** was the anomalous driving exhibited
  - ***Real-time*** application ; a pilot version already deployed by ***Uber***

Applications

- **Rating cab drivers :**
  - ***GPS trackers*** already installed in all cabs
  - Identifying ***how*** anomalous and ***where*** was the anomalous driving exhibited
  - ***Real-time*** application ; a pilot version already deployed by ***Uber***
- **Personalized Car Insurance**, ***Pay How You Drive (PHYD)***, ***Usage Based Insurance (UBI)***
  - Direct approach to ***assess driving behaviour*** from user's historical driving records
  - ***Mutually beneficial*** scheme ; for the insurance company and the driver
  - Efforts steered in ***USA, Japan, Australia, EU*** partnered with ***Toyota***

Notations

- **Road Network** : Directed graph $G(V, E)$; set of nodes V and set of edges E
- **D** denote prevailing traffic conditions containing trajectories in history

Notations

- **Road Network** : Directed graph $G(V, E)$; set of nodes V and set of edges E
- **D** denote prevailing traffic conditions containing trajectories in history
- **Trajectory**

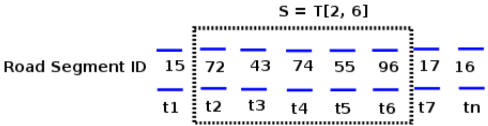
Road Segment ID	15	72	43	74	55	96	17	16
	t1	t2	t3	t4	t5	t6	t7	tn

Notations

- **Road Network** : Directed graph $G(V, E)$; set of nodes V and set of edges E
- **D** denote prevailing traffic conditions containing trajectories in history
- **Trajectory**
- Time taken to traverse T $times(T) = \sum_{\forall e \in S} T.e_t$

Notations

- **Road Network** : Directed graph $G(V, E)$; set of nodes V and set of edges E
- **D** denote prevailing traffic conditions containing trajectories in history
- **Trajectory**
- Time taken to traverse T $times(T) = \sum_{\forall e \in S} T.e_t$
- **Sub-trajectory**



Notations

- **Road Network** : Directed graph $G(V, E)$; set of nodes V and set of edges E
- **D** denote prevailing traffic conditions containing trajectories in history
- **Trajectory**
- Time taken to traverse T $times(T) = \sum_{\forall e \in S} T.e_t$
- **Sub-trajectory**
- **Maximal Anomalous Sub-Trajectory (MAS)** S : no *anomalous super-trajectory* of S

PROBLEM STATEMENT

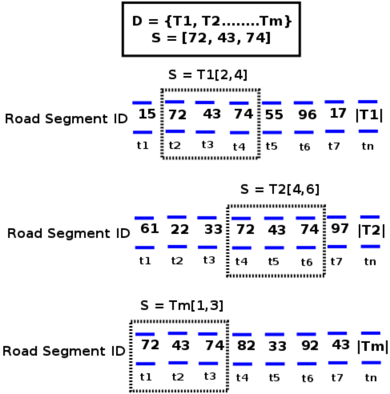
Given $\mathbb{D}$, the reference dataset of trajectories. For an input trajectory, identify all of its maximal temporally anomalous sub-trajectories under a user-provided threshold θ with respect to $\mathbb{D}$.

Anomaly Model

- Standard **z-score** based anomaly model with *Normal Distribution*
- Distribution of travel times along $S = N(\mu_S, \sigma_S^2)$, where μ_S is the **mean** and σ_S^2 is the **variance**

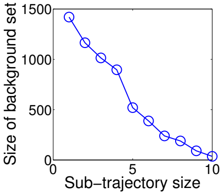
Anomaly Model

- Standard **z-score** based anomaly model with *Normal Distribution*
- Distribution of travel times along $S = N(\mu_S, \sigma_S^2)$, where μ_S is the **mean** and σ_S^2 is the **variance**
- Compute μ_S and σ_S^2 from the background set of S in $\mathbb{D}$



Anomaly Model

- Standard **z-score** based anomaly model with *Normal Distribution*
- Distribution of travel times along $S = N(\mu_S, \sigma_S^2)$, where μ_S is the **mean** and σ_S^2 is the **variance**
- Compute μ_S and σ_S^2 from the background set of S in $\mathbb{D}$
- Issue of **data sparsity** ; non existent background set for sub-trajectory size > 10

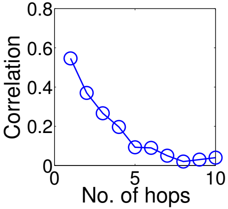


Managing Data Sparsity

- Covariance $cov(e, e')$ captures the dependence between travel times of the two edges e and e'

Managing Data Sparsity

- Covariance $cov(e, e')$ captures the dependence between travel times of the two edges e and e'



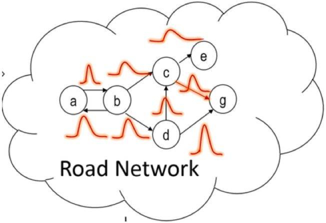
1. $\forall e, e' \in E$ more than 5 hops away, $cov(e, e') \approx 0$
2. $\forall e, e' \in E$, $cov(e, e') \geq 0$

Managing Data Sparsity

- Covariance $cov(e, e')$ captures the dependence between travel times of the two edges e and e'

  1. $\forall e, e' \in E$ more than 5 hops away, $cov(e, e') \approx 0$
  2. $\forall e, e' \in E$, $cov(e, e') \geq 0$

- For $\forall e \in E$, fit a normal distribution $time(e) = \mathcal{N}(\mu_e, \sigma_e^2)$



Managing Data Sparsity

- Covariance $cov(e, e')$ captures the dependence between travel times of the two edges e and e'

1. $\forall e, e' \in E$ more than 5 hops away, $cov(e, e') \approx 0$
2. $\forall e, e' \in E$, $cov(e, e') \geq 0$

- For $\forall e \in E$, fit a normal distribution $time(e) = \mathcal{N}(\mu_e, \sigma_e^2)$

- Modelling travel times along S as **multivariate distribution** of its edges

$$\sigma_S^2 = \sum_{\forall e \in S} \sigma_e^2 + 2 \sum_{\forall \{e, e'\} \in S} cov(e, e') \quad (1)$$

Anomaly Model

- **Deviation of** S as an *aggregate* of the deviation in its constituent edges

$I(e)(\mu_e - t_e)^2$	0	0.5	1.5	2	0.5	-1	-1.5	-2.5
Road Segment	1	2	3	4	5	6	7	8

MAS 1 MAS 2

Anomaly Model

- **Deviation of S** as an *aggregate* of the deviation in its constituent edges

$I(e)(\mu_e - t_e)^2$	0	0.5	1.5	2	0.5	-1	-1.5	-2.5
Road Segment	1	2	3	4	5	6	7	8

MAS 1 (segments 2–5) MAS 2 (segments 5–8)

-

$$\textbf{dist(S)} = \sum_{\forall e \in S} \mathcal{I}(e)(\mu_e - S.t_e)^2 \tag{2}$$

$$\mathcal{I}(e) = \begin{cases} 1 & \text{if } S.t_e >= \mu_e \textbf{ : Over-speeding} \\ -1 & \text{if } S.t_e < \mu_e \textbf{ : Under-speeding} \end{cases} \tag{3}$$

Anomaly Model

- **Deviation of S** as an *aggregate* of the deviation in its constituent edges
- $$\textbf{dist(S)} = \sum_{\forall e \in S} \mathcal{I}(e)(\mu_e - S.t_e)^2 \quad (2)$$

$$\mathcal{I}(e) = \begin{cases} 1 & \text{if } S.t_e >= \mu_e \textbf{ : Over-speeding} \\ -1 & \text{if } S.t_e < \mu_e \textbf{ : Under-speeding} \end{cases} \quad (3)$$

- S is **anomalous** if

$$\frac{|dist(S)|}{\sigma_S^2} > \theta \quad (4)$$

Anomaly Model

- **Deviation of S** as an *aggregate* of the deviation in its constituent edges
-

$$\textbf{dist(S)} = \sum_{\forall e \in S} \mathcal{I}(e)(\mu_e - S.t_e)^2 \tag{2}$$

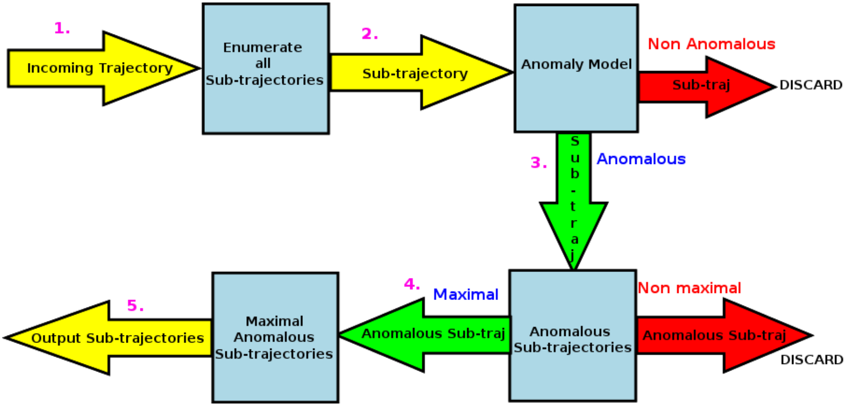
$$\mathcal{I}(e) = \begin{cases} 1 & \text{if } S.t_e >= \mu_e \textbf{ : Over-speeding} \\ -1 & \text{if } S.t_e < \mu_e \textbf{ : Under-speeding} \end{cases} \tag{3}$$

- S is **anomalous** if

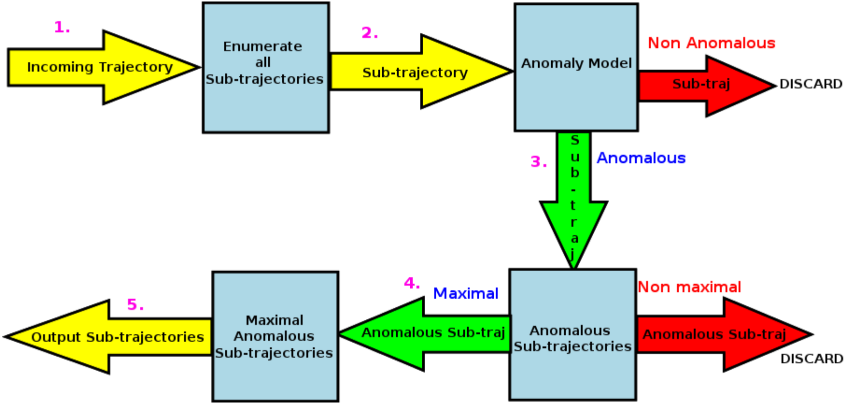
$$\frac{|dist(S)|}{\sigma_S^2} > \theta \tag{4}$$

- Anomalous sub-trajectory **can** contain **non-anomalous edges**.
- Anomalous sub-trajectory **must** contain at least one **anomalous edge**.

Approach 1 : The Naïve Approach



Approach 1 : The Naïve Approach



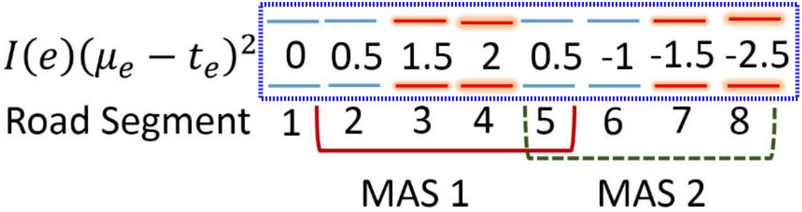
- Computation complexity of T with n edges = $\mathbb{O}(n^2)$; not scalable

Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**

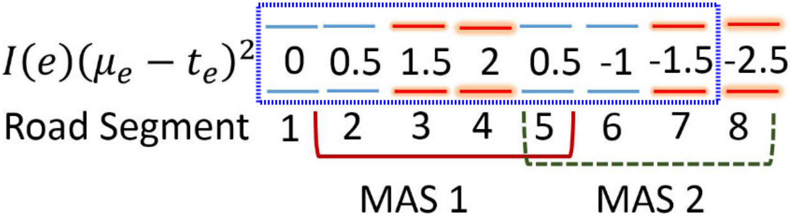
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



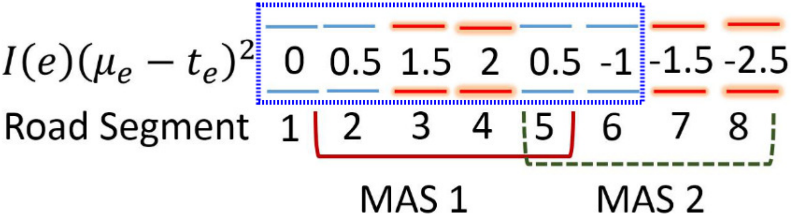
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



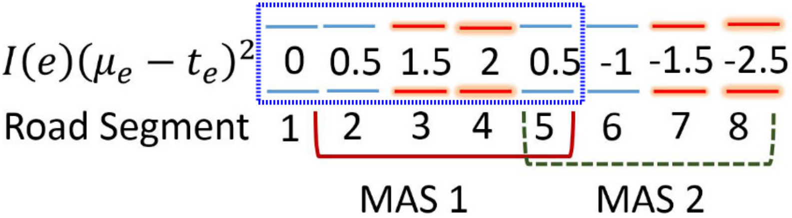
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



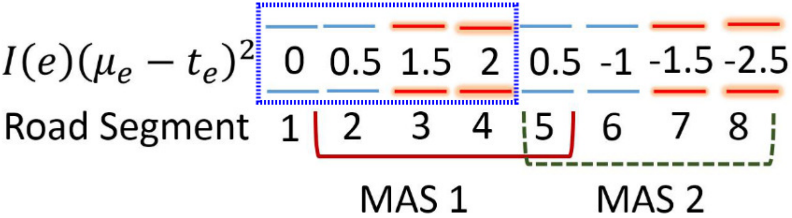
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



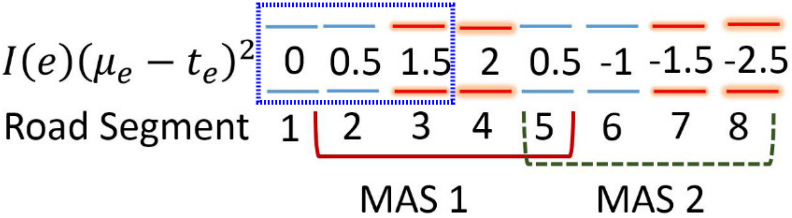
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



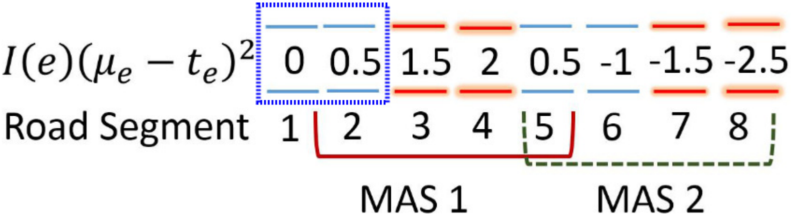
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



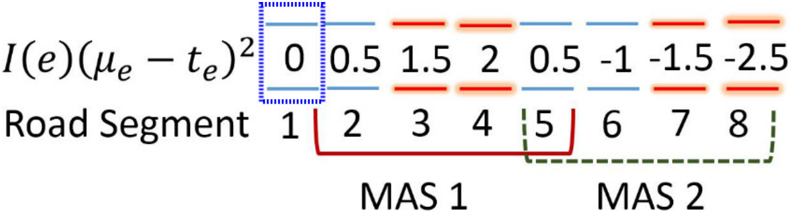
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



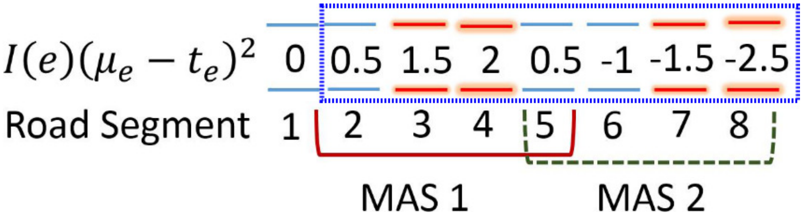
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



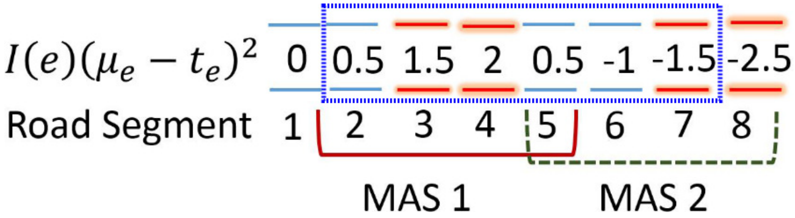
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



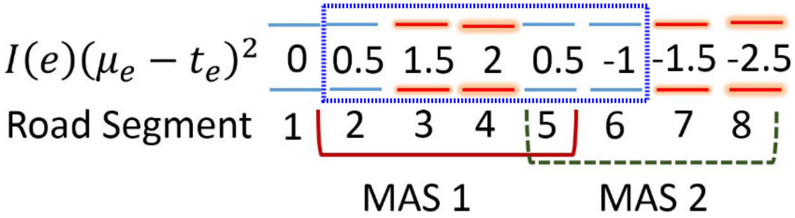
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



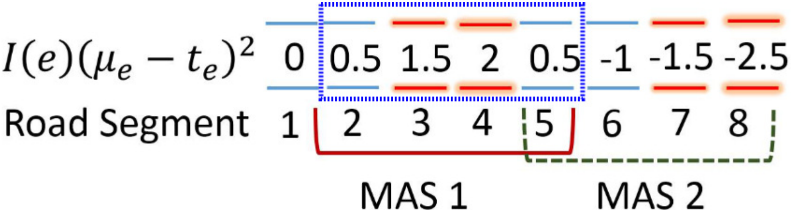
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



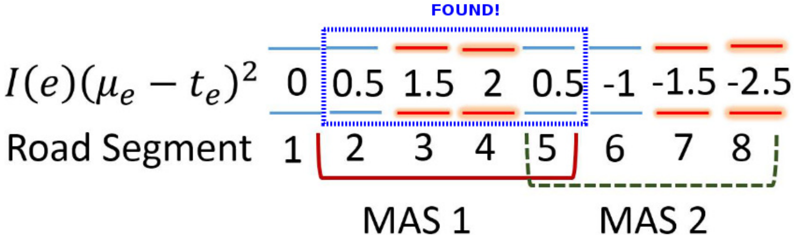
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



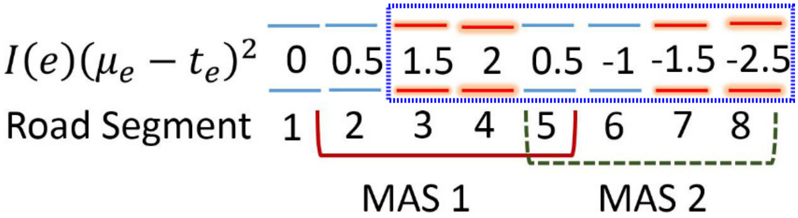
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



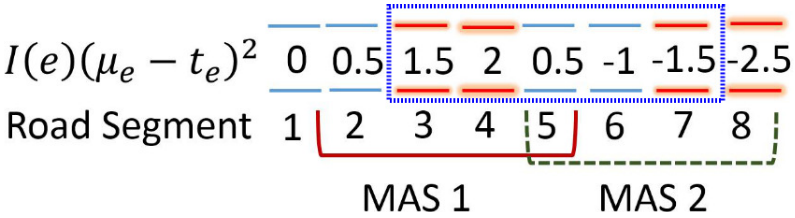
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



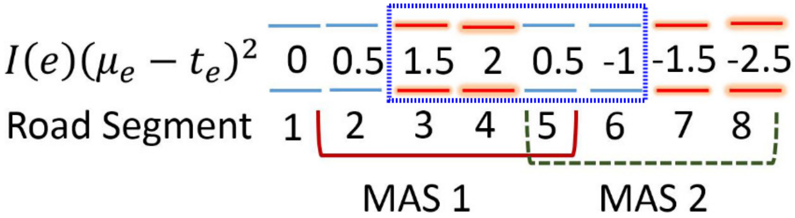
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



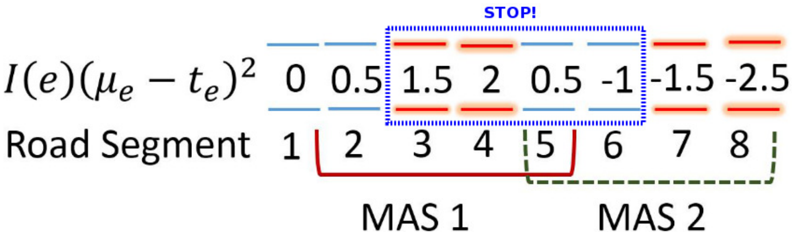
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



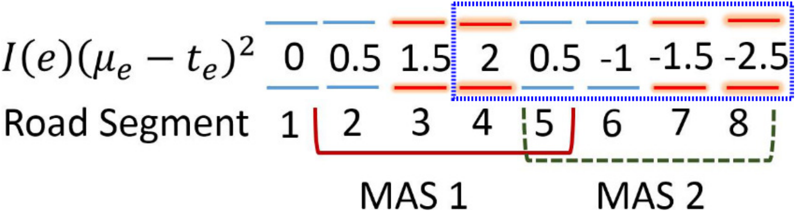
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



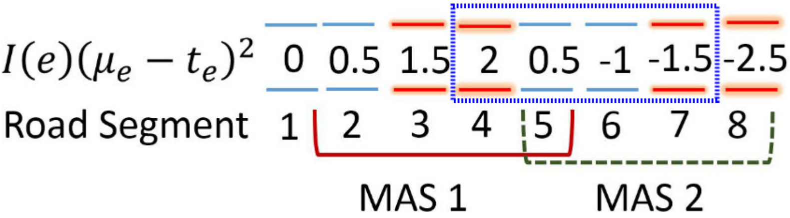
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



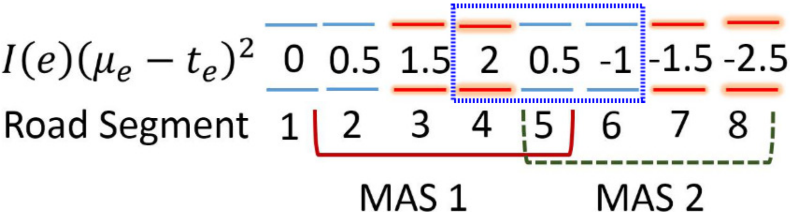
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



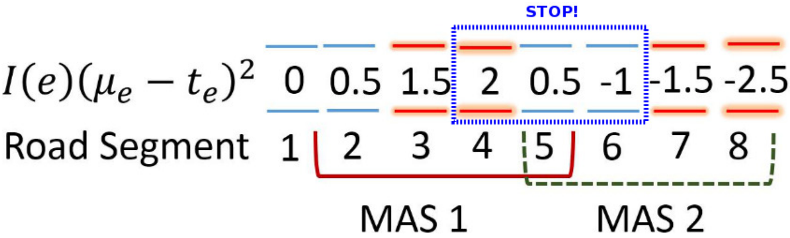
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



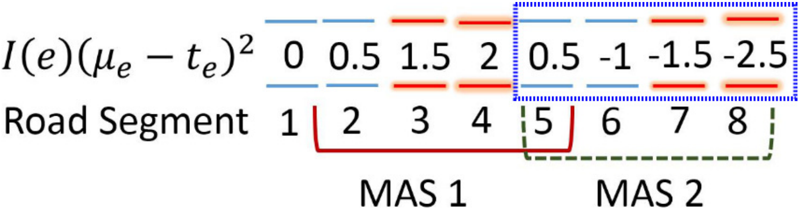
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



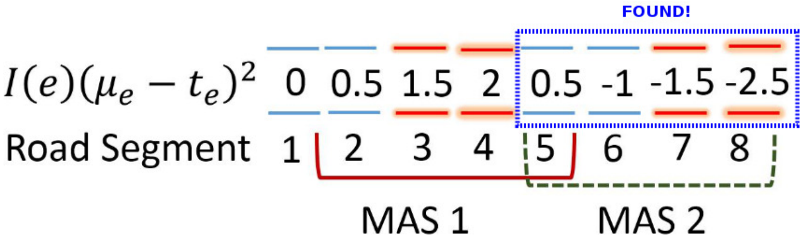
Approach 2 : Bi-Directional Sliding Window

- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**

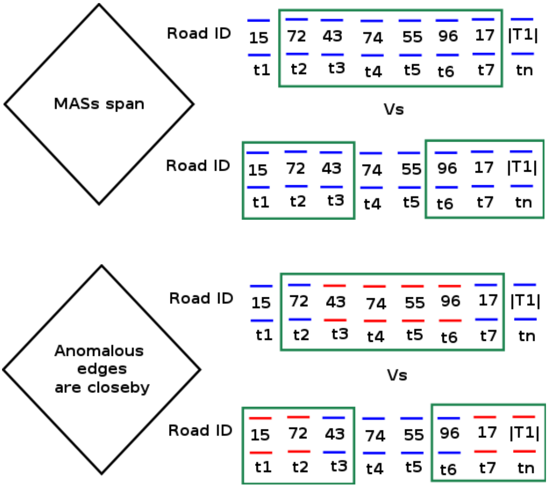


Approach 2 : Bi-Directional Sliding Window

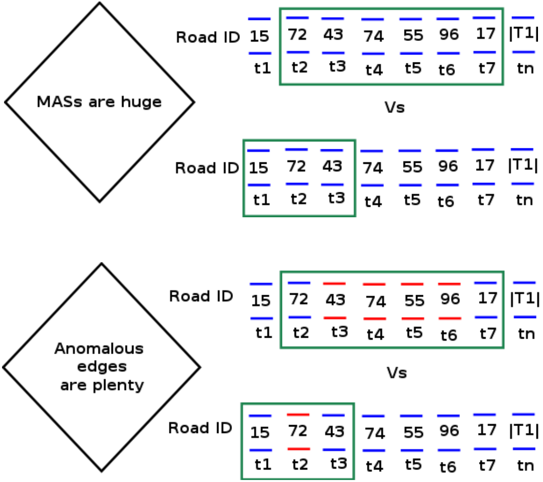
- **Avoid** evaluating **non-maximal** anomalous sub-trajectories
- Evaluating **longer sub-trajectories first**



1. Best scenario for Bi-Directional Sliding Window



2. Best scenario for Bi-Directional Sliding Window



MANTRA's Mantra

MANTRA identifies segments of input trajectory which are best suited for Bi Directional Sliding Window.

MANTRA applies BSW on these special segments, called *Islands*.

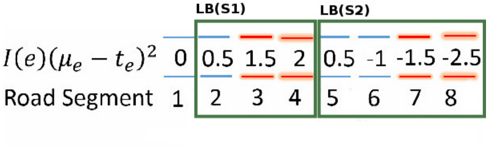
Impact Regions

- **Seeds (contiguous anomalous edges) :** $S_1 = T[3:4]$ and $S_2 = T[7:8]$

$I(e)(\mu_e - t_e)^2$	0	0.5	1.5	2	0.5	-1	-1.5	-2.5
Road Segment	1	2	3	4	5	6	7	8

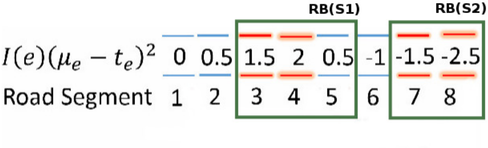
Impact Regions

- **Seeds (contiguous anomalous edges) :** $S_1 = T[3:4]$ and $S_2 = T[7:8]$
- **Left Boundary :**



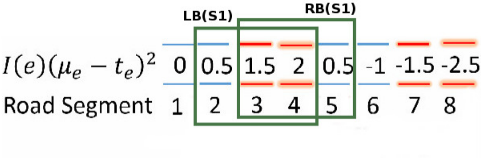
Impact Regions

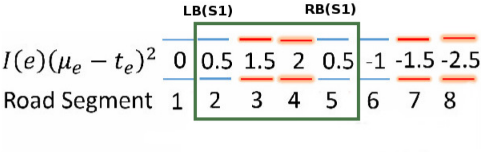
- **Seeds (contiguous anomalous edges) :** $S_1 = T[3:4]$ and $S_2 = T[7:8]$
- **Left Boundary :**
- **Right Boundary :**



Continued

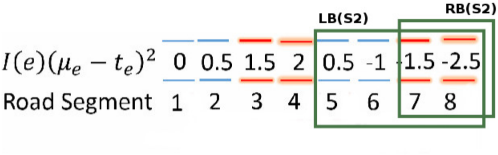
- **Impact Region :**
  - S_1

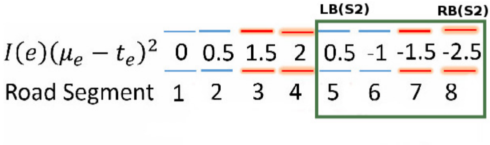




Continued

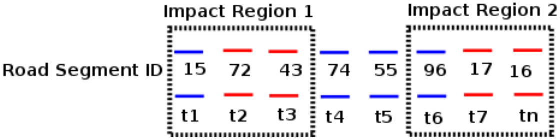
- **Impact Region :**
  - S_1
  - S_2





Impact Regions

Impact Regions separated by more than one non-anomalous edges DO NOT interact.



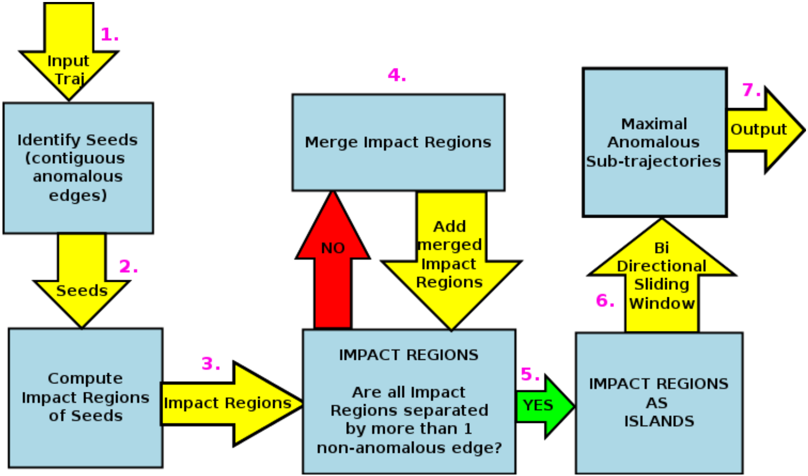
Such Impact Regions are called *Islands*.

All MASs are contained within *Islands* and do not span across them.

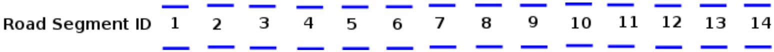
***Islands* are best suited for Bi Directional Sliding Window.**

MANTRA Pipeline

APPROACH 3. MANTRA



MANTRA Example



MANTRA Example

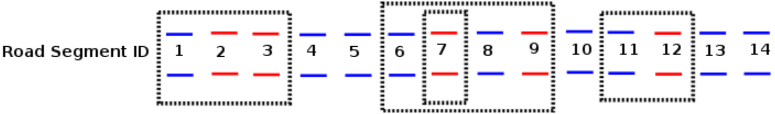
- **Seed identification**

Road Segment ID 1 2 3 4 5 6 7 8 9 10 11 12 13 14

MANTRA Example

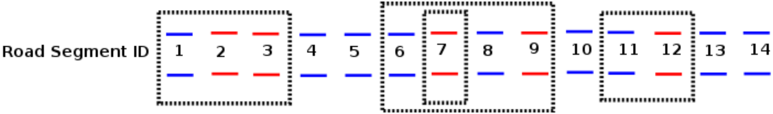
- Seed identification

- **Compute impact region of the seeds**



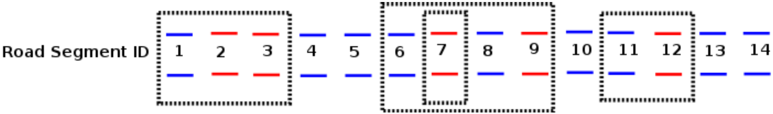
Continued

- **Merge interacting impact regions of seeds till convergence**

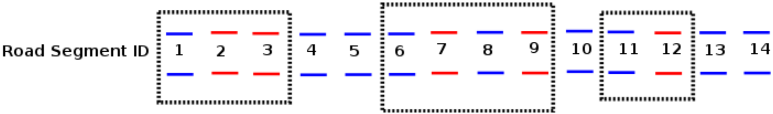


Continued

- **Merge interacting impact regions of seeds till convergence**

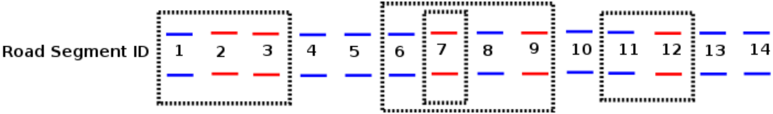


- 1^{st} iteration :

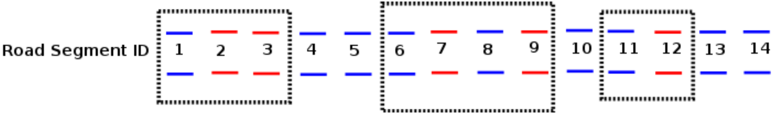


Continued

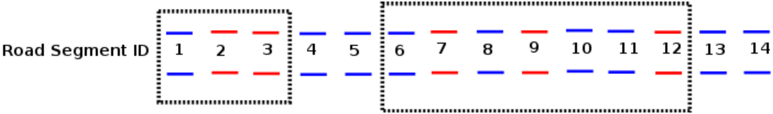
- **Merge interacting impact regions of seeds till convergence**



- 1^{st} iteration :

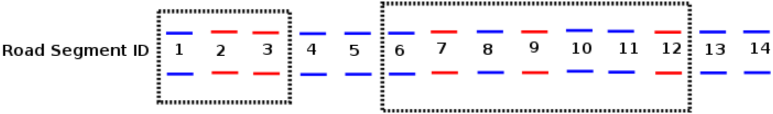


- 2^{nd} iteration :



Continued

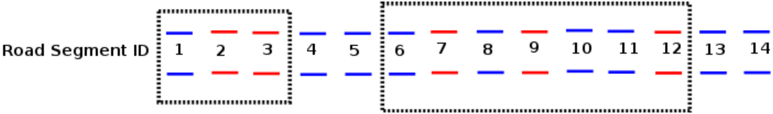
- **Merge interacting impact regions of seeds till convergence**
- 1^{st} iteration :
- 2^{nd} iteration :



- **Final seeds** : $ST1[1:3]$ and $ST5[6:12]$

Continued

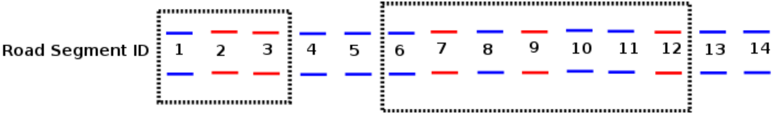
- Merge interacting impact regions of seeds till convergence
- 1^{st} iteration :
- 2^{nd} iteration :



- **Final seeds** : $ST1[1:3]$ and $ST5[6:12]$
- **Islands : non-interacting impact regions**

Continued

- **Merge interacting impact regions of seeds till convergence**
- 1^{st} iteration :
- 2^{nd} iteration :



- **Final seeds** : $ST1[1:3]$ and $ST5[6:12]$
- **Islands : non-interacting impact regions**
- **Perform Bi-Directional Sliding Window on the islands**

Set Up and Datasets

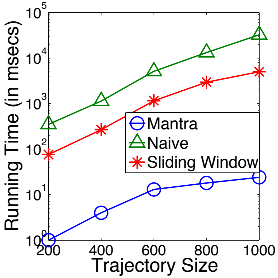
EXPERIMENTATION SET UP

- Java JDK 1.7.0
- 12GB memory
- Intel i5 2.60GHz quad core processor
- Ubuntu 13.04

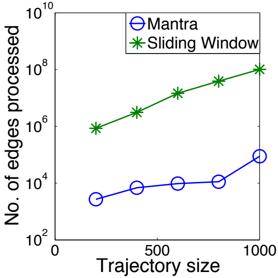
DATASETS FROM BEIJING

- T-drive dataset
  - Largest publicly available trajectory dataset
  - 136,759 trajectories
- Geolife dataset
  - 18760 trajectories
  - Vehicle annotated trajectories : car, walk, bus

Effect of Trajectory Size



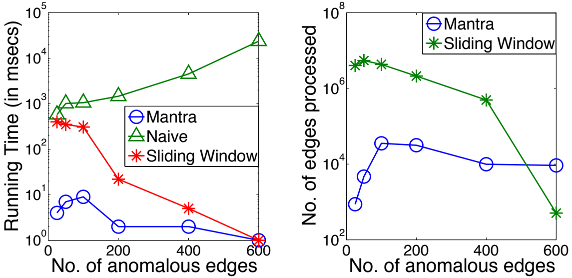
(a)



(b)

1. MANTRA is upto **3 orders of magnitude faster** ; **less** number of edges processed
2. For longer trajectories, MANTRA **consumes $<$ 25 ms**

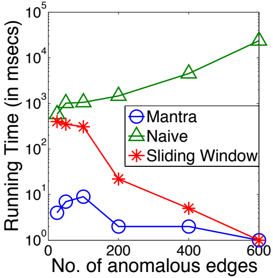
Effect of Number of Anomalous Edges



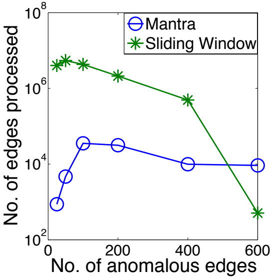
(c) (d)

1. For sliding window, **number of anomalous edges ↑ $\Longrightarrow$ running time ↓, # edges processed ↓**
2. For MANTRA, **hump $\widehat{=}$ more iterations to converge** islands formation

Effect of Number of Anomalous Edges



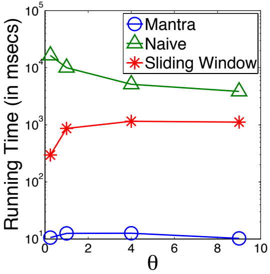
(e)



(f)

1. For sliding window, **number of anomalous edges ↑ $\implies$ running time ↓, # edges processed ↓**
2. For MANTRA, **hump** $\widehat{=}$ **more iterations to converge** islands formation
3. Sliding Window overtakes MANTRA ;**islands formation redundant with ↑ in anomalous edges**

Effect of Anomaly Threshold on Runtime



- **$\uparrow$ threshold $\implies$ $\downarrow$ anomalous edges**
- Running time for Naive $\downarrow$ with $\uparrow$ in threshold ; less anomalous sub-trajectories
- **Running time Sliding Window $\uparrow$ with $\uparrow$ in threshold** ; less anomalous edges
- **Hump $\widehat{=}$ convergence of island formation**

Efficacy and Applications

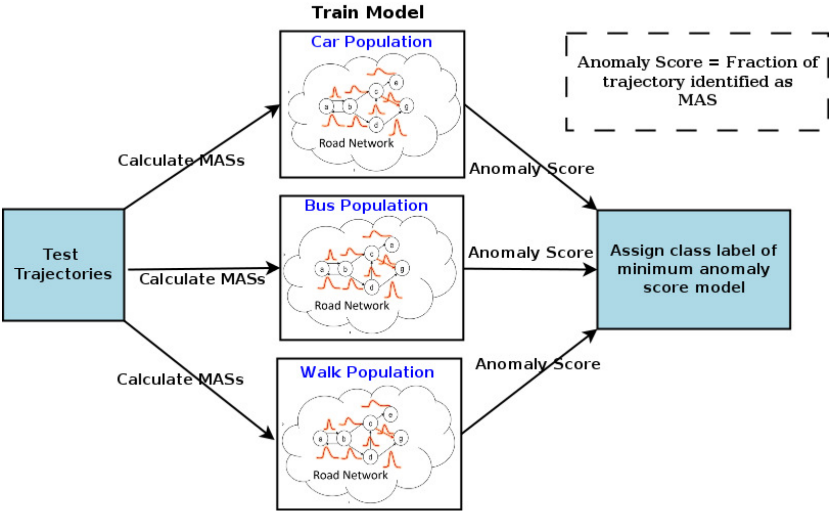
- ***Are we able to identify sub-trajectories that would be considered anomalous by humans ?***

Efficacy and Applications

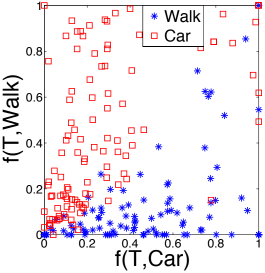
- ***Are we able to identify sub-trajectories that would be considered anomalous by humans ?***

- **Intuition** :
  - *Car* trajectory **least** anomalous against *Car* population
  - *Car* trajectory **more** anomalous against *Walk* and *Bus* population

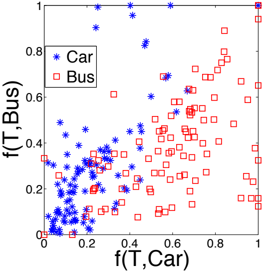
Trajectory Classification : Classification Model



Trajectory Classification

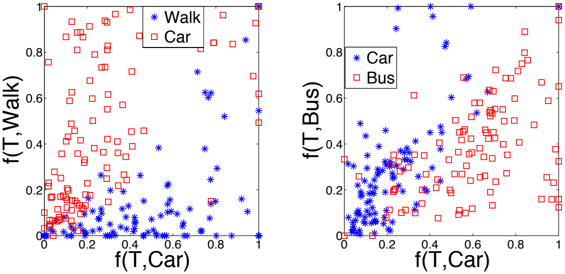


(g) Walk and Car



(h) Bus and Car

Trajectory Classification



(i) Walk and Car

(j) Bus and Car

- Majority *Walk* trajectories have **higher anomaly score** against *Car*
- *Car* vs *Walk* is **more drastic** than *Car* vs *Bus*

Trajectory Classification

(k) Walk and Car

(l) Bus and Car

- Majority *Walk* trajectories have **higher anomaly score** against *Car*
- *Car* vs *Walk* is **more drastic** than *Car* vs *Bus*
- Two class classification f-score

Class label	Walk	Bus
Car	0.85	0.74
Walk	-	0.75

Trajectory Classification

(m) Walk and Car

(n) Bus and Car

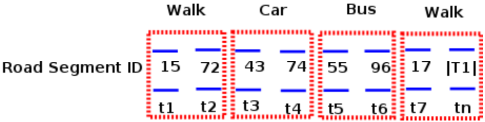
- Majority *Walk* trajectories have **higher anomaly score** against *Car*
- *Car* vs *Walk* is **more drastic** than *Car* vs *Bus*
- Two class classification f-score

Class label	Walk	Bus
Car	0.85	0.74
Walk	-	0.75

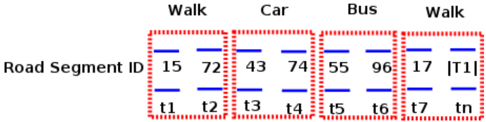
- Three class classification f-score

Class combination	car	walk	bus
Car-Walk-Bus	0.62	0.69	0.47

Trajectory Segmentation

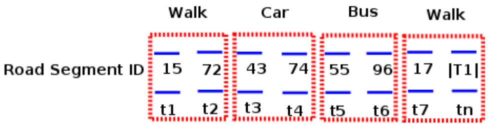


Trajectory Segmentation



- Segmentation model :
  - Identify MASs on test trajectory against each of *Walk*, *Car*, *Bus* model
  - Assign class labels to **MASs** edges to the **closest class** ; i.e the model it is **least anomalous against**

Trajectory Segmentation



- Segmentation model :
  - Identify MASs on test trajectory against each of *Walk*, *Car*, *Bus* model
  - Assign class labels to **MASs** edges to the **closest class** ; i.e the model it is **least anomalous against**
- F-score based on number of edges identified correctly

Class label	Walk	Bus
Car	0.80	0.65
Walk -		0.76

Conclusions

- Unique problem of mining **Maximal Anomalous Sub-trajectories**
- MANTRA refines the search space and identifies ***islands*** where all the MASs are present
- MANTRA is observed to be **3** orders of magnitude faster than baseline
- MANTRA conforms to human intuition of anomaly demonstrated through ***classification*** and ***segmentation***
- MANTRA facilitates a **unique tool to classify segments of trajectories** based on vehicle type from their GPS traces