

Periodic patterns in human mobility

Matthew James Williams

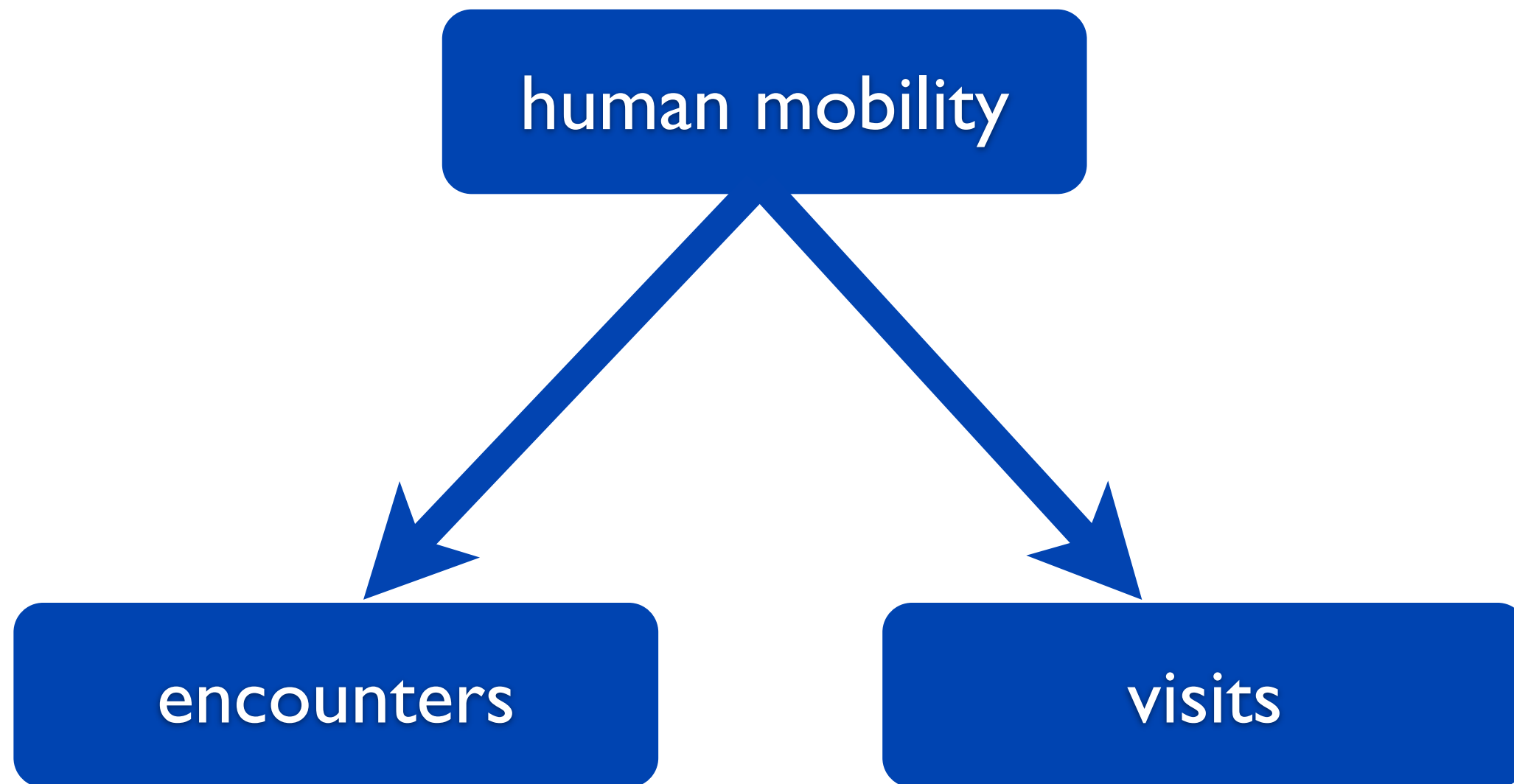
Cardiff University
School of Computer Science
& Informatics
United Kingdom

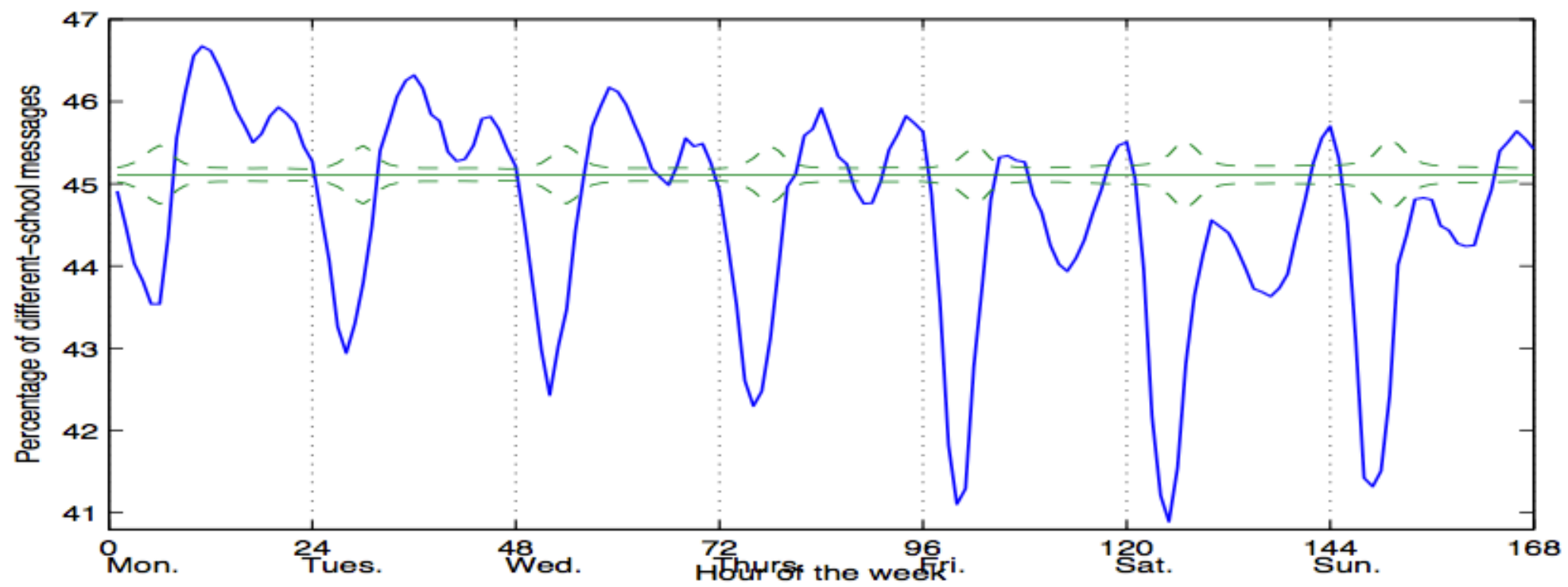


VLunch Seminar
3rd October 2013

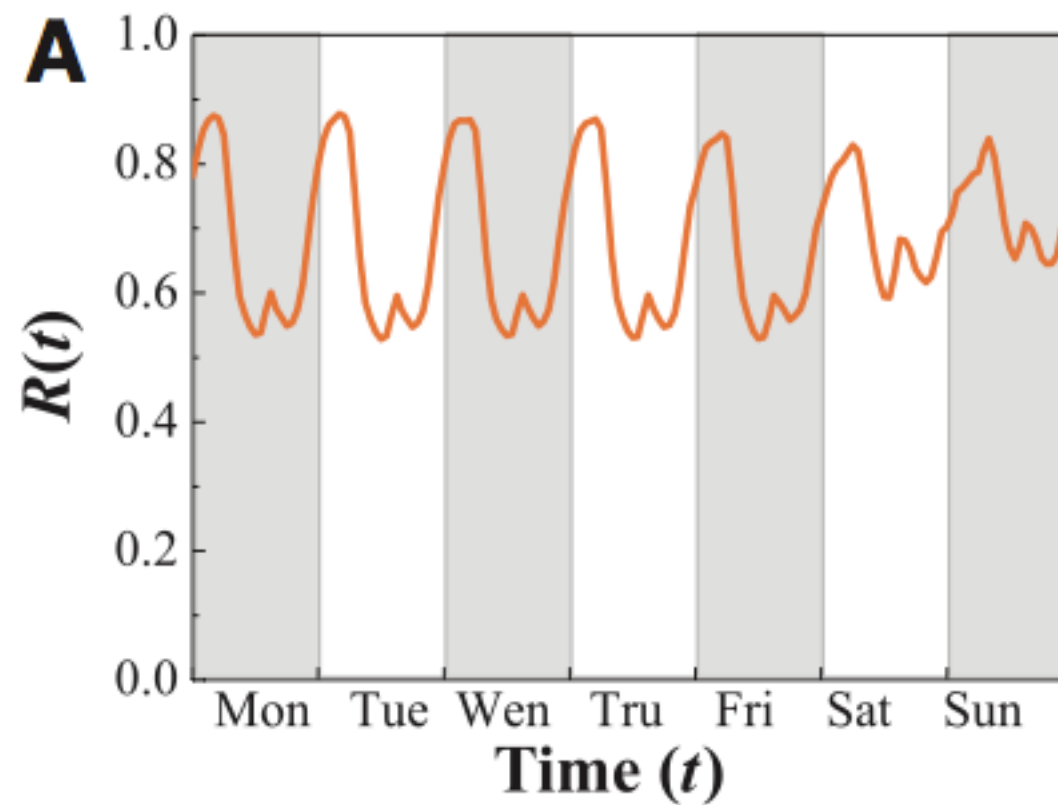
Introduction

Can we quantify and exploit periodicity in individuals' mobility patterns?

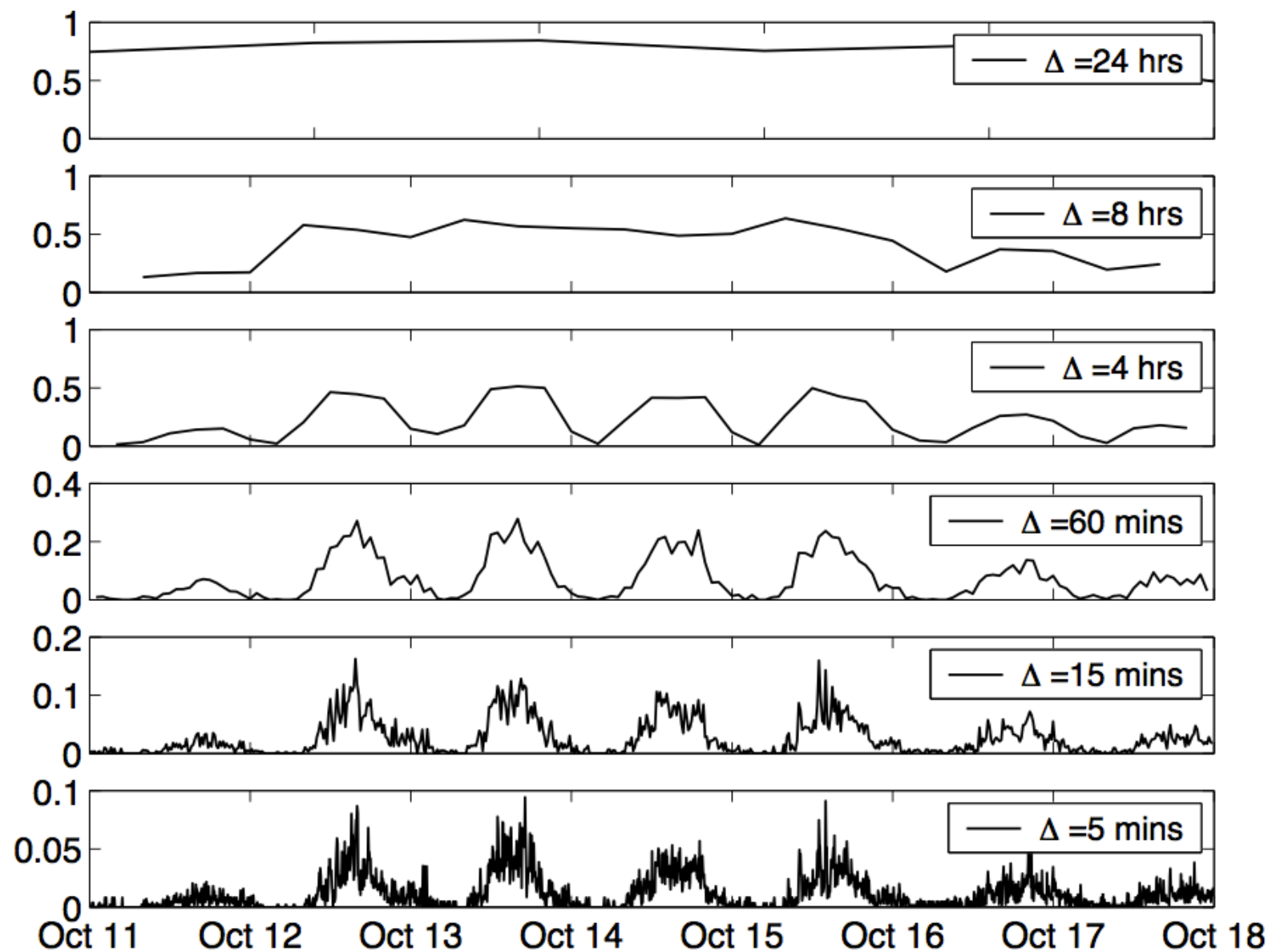




Facebook messaging rates



likelihood of individual being
at most-visited location



**network connectivity of
Bluetooth encounters**



routine in human mobility gives rise to regular mobility behaviour

identifying regular mobility has many possible applications



human-based
opportunistic networks



context for
digital assistants



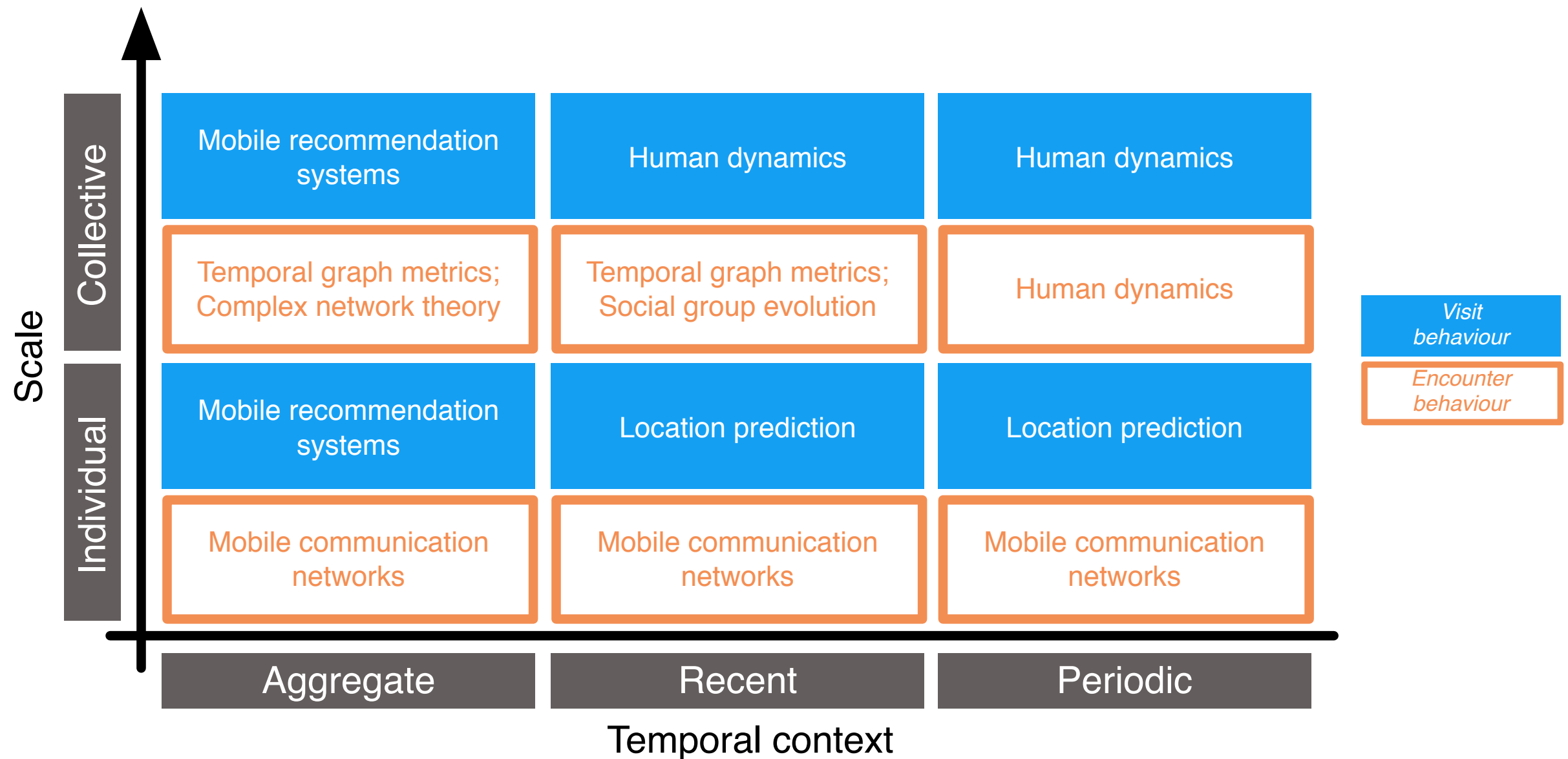
personalised customer
service

...and more

Opportunistic networks

- **Opportunistic networks (oppnets)** are a broad class of networks where messages are spread by the **mobility of individuals** and their occasional **physical encounters**
- **Encounters** are the fundamental unit of communication in these networks
- Modelling temporal context in forwarding decisions has resulted in improved content-sharing performance

Who's interested?



Objective & scope

Exploring the presence and character of periodic patterns in the visits and encounters of human individuals for use as context in a variety of decentralised context-aware applications by proposing methods that operate on an event stream representation of data.

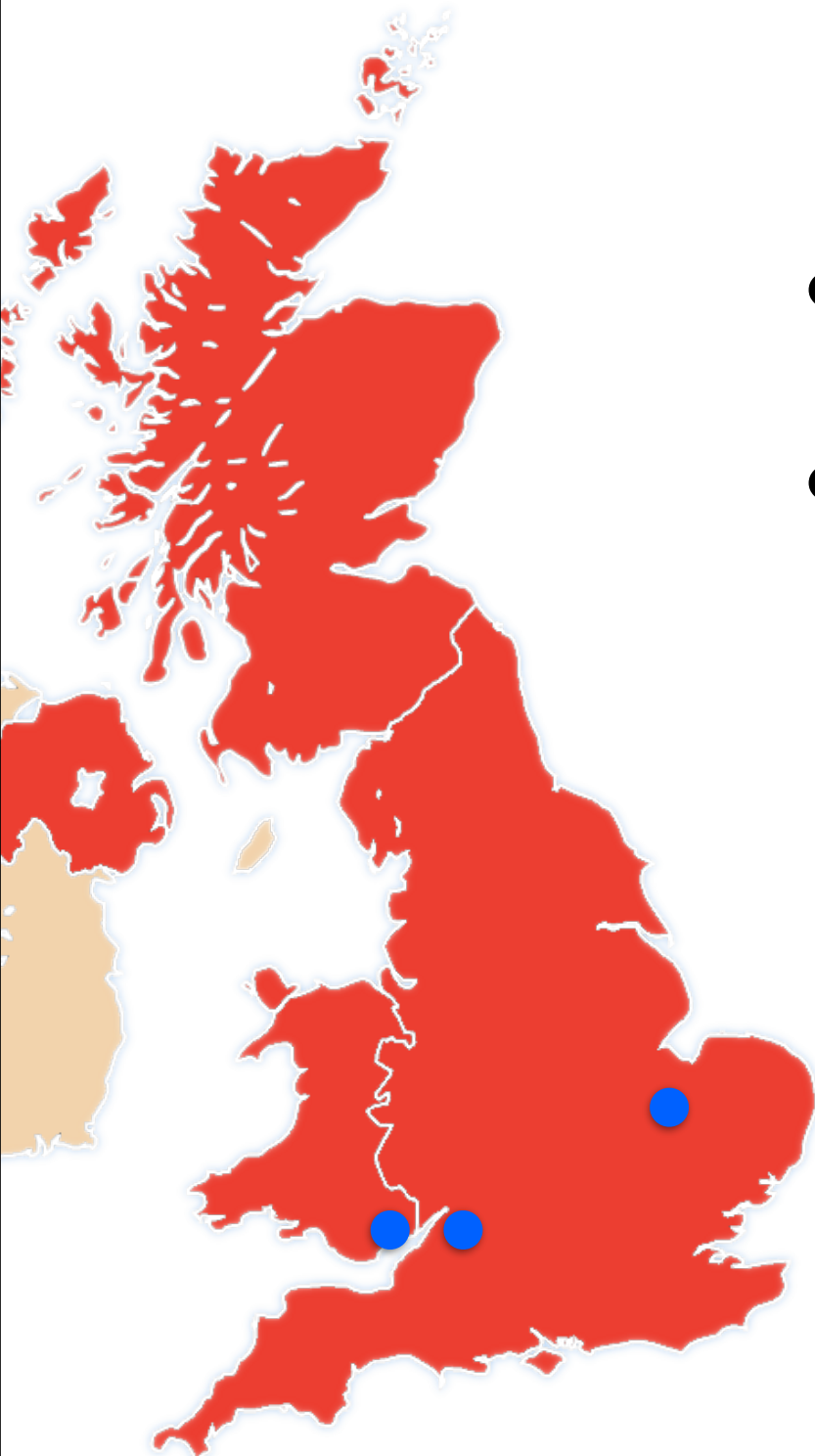
- Key points:
 - Periodic patterns
 - Individual context
 - Decentralised methods
 - Event stream data

Overview

- **Datasets**
- **Part 1 – Visits:**
Approach borrowed from spike train analysis (neuroscience) to measure regularity in event data
- **Part 2 – Encounters:**
Data mining approach for identifying periodic encounter community behaviour
Spike train approach to periodic encounter community detection
- **Future work**

Datasets

Foursquare venue checkins



- Checkins to venues on Foursquare
- Checkins collected for three urban areas in the UK: Cardiff, Cambridge, and Bristol
- Locations: Foursquare venues
- Users: Foursquare users



London Underground Stations



- Visits to London Underground stations recorded by the Oyster card automated fare collection system
- Locations: London Underground stations
- Users: passengers using the Oyster card system
- Includes ~80 million journeys made during 28 days



Dartmouth WLAN APs



- WLAN accesses on Dartmouth college campus (USA)
- Locations: access points
- Users represented by devices carried by staff and students
- Majority of devices are laptops, as this dataset is from 2004
- Visits to APs
- Encounters when two individuals at same AP



Reality Mining Bluetooth Encounters



- 100 MIT students given smartphones with Bluetooth encounter logging software (they were aware!)
- Tracked for 9 months during 2004 and 2005
- Logged ~7 million encounters among the 100 students



Visits

	FOURSQUARE	DARTMOUTH	UNDERGROUND
Area(s)	Bristol, Cardiff, & Camb.	Dartmouth	London
Scale	Urban	Campus	Metropolitan
Duration	54 days	365 days	28 days
Location type	Venue	Access point	Metro station
Visit type	Checkin	Association	Card swipe
Individuals	5,654	7,187	5,177,134
Locations	4,774	567	270
Visits	43,767	4,724,255	159,111,519
Visits per day	811	12,943	5,682,554
Visits per individual per day	0.14	1.80	1.10
M	16,155	275,843	35,338,486
$\langle L \rangle$	2.7	17.0	4.5

Table 3.2: Summary of base visit datasets. M denotes the number of chronologies and $\langle L \rangle$ denotes the mean number of visits per chronology. A chronology $S_{v,l}$ is only included in a dataset if v visited l at least once in the duration of the dataset. Only active individuals and locations are counted; that is, locations and individuals are only counted if they were involved in at least one chronology.

Encounters

	DARTMOUTH	REALITY
Area(s)	Dartmouth	MIT
Duration	365 days	270 days
Encounter type	Access point colocation	Bluetooth proximity
Encounter range	$\leq 40\text{m}$	$\leq 10\text{m}$
Individuals	7,173	100
Encounters	6,800,755	531,703
Encounters per day	18,632.2	1,969.3
Encounters per individual per day	2.6	19.7
M	897,996	2,675
$\langle L \rangle$	7.6	198.8

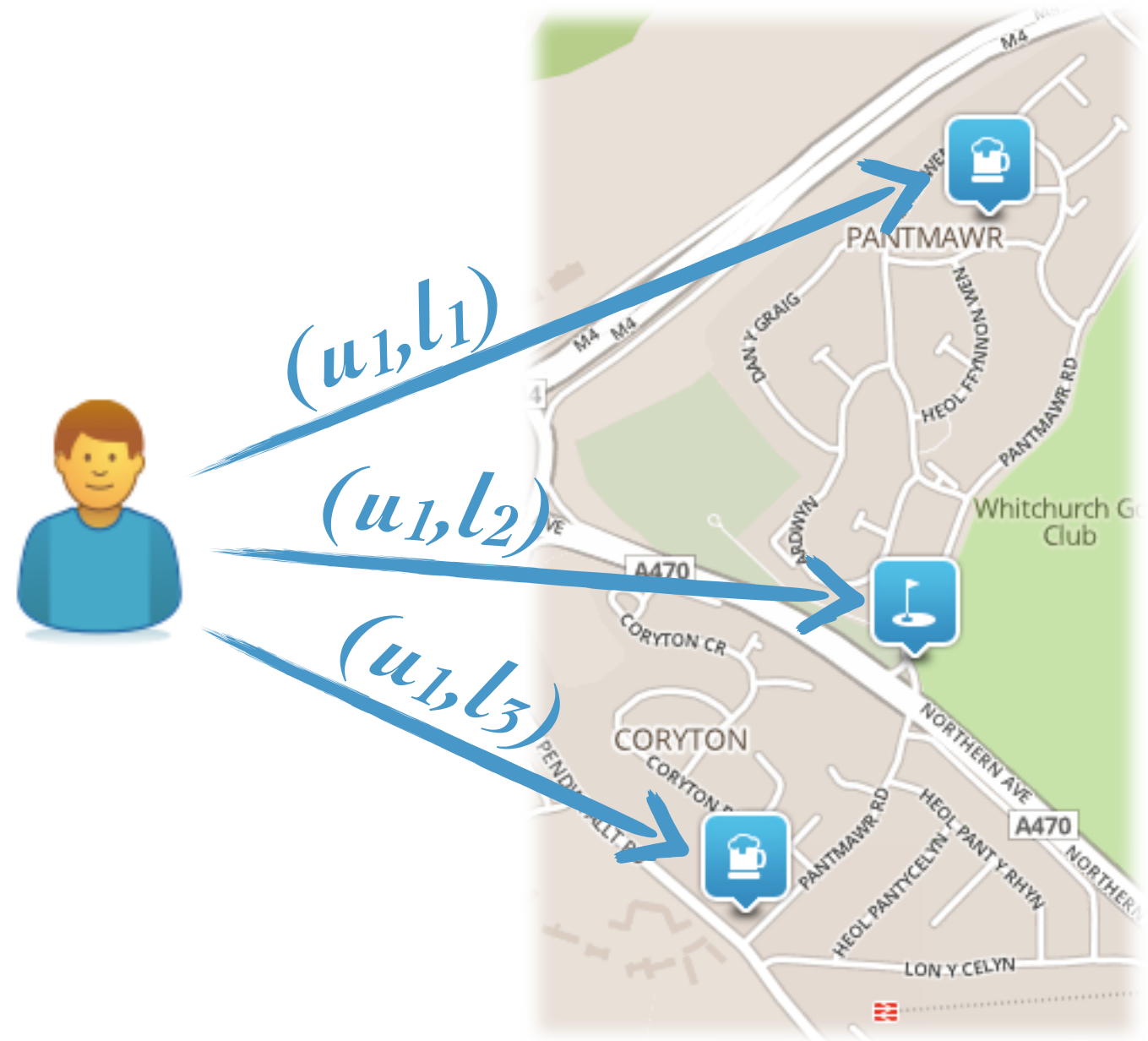
Table 3.3: Summary of base encounter datasets. M denotes the number of chronologies and $\langle L \rangle$ denotes the mean number of encounters per chronology. A chronology $\mathcal{S}_{v,u}$ is only included in a dataset if v encountered u at least once in the duration of the dataset. Only active individuals are counted; that is, an individual is only counted if he/she was involved in at least one chronology.

Part I

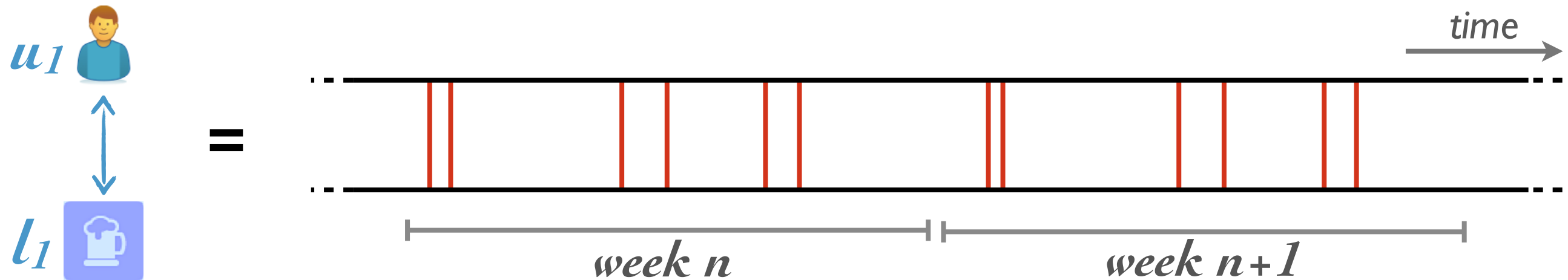
Measuring periodicity in individual visiting patterns

user-at-location chronologies

We call the history of visits for a particular user u at a particular location l a visit chronology



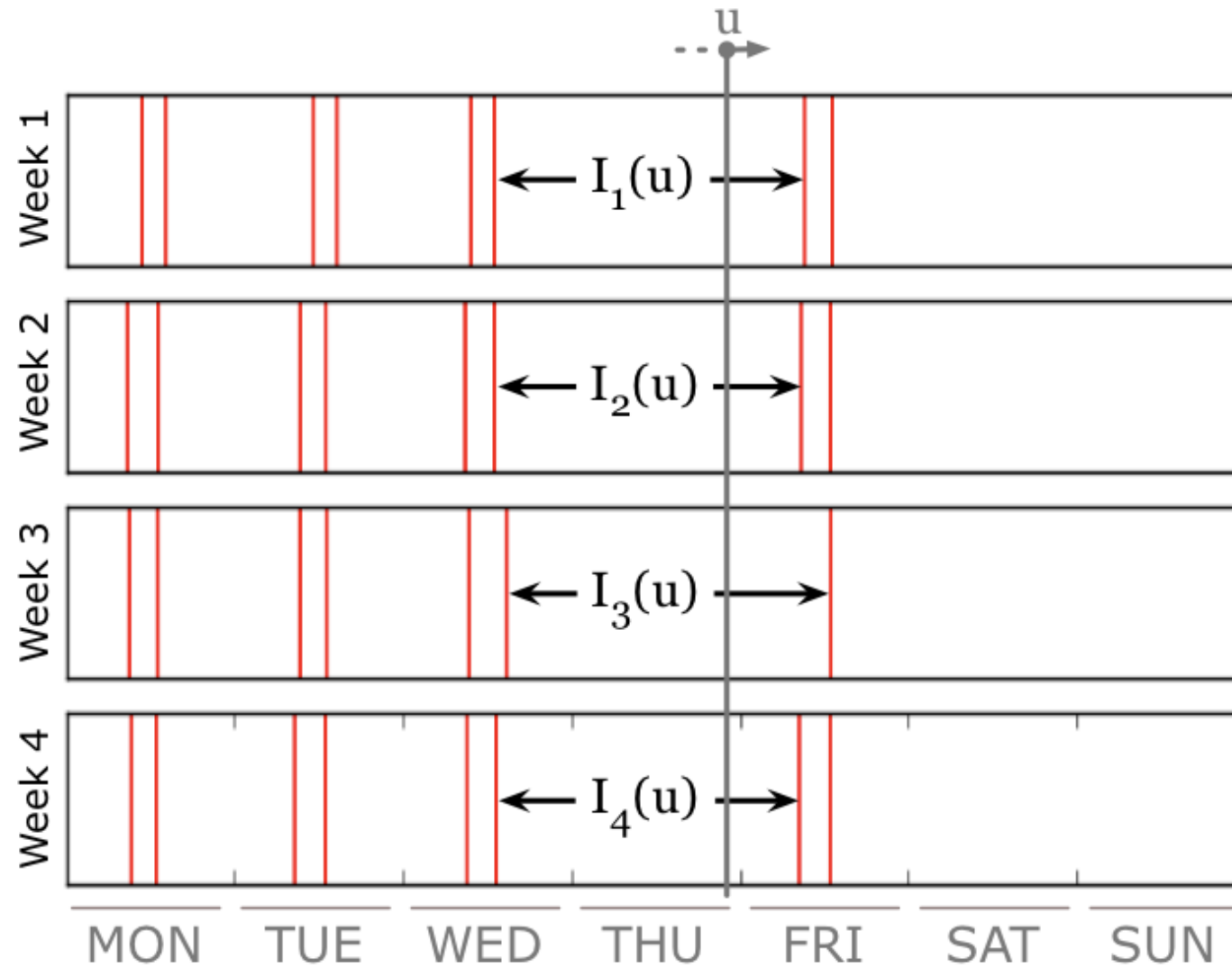
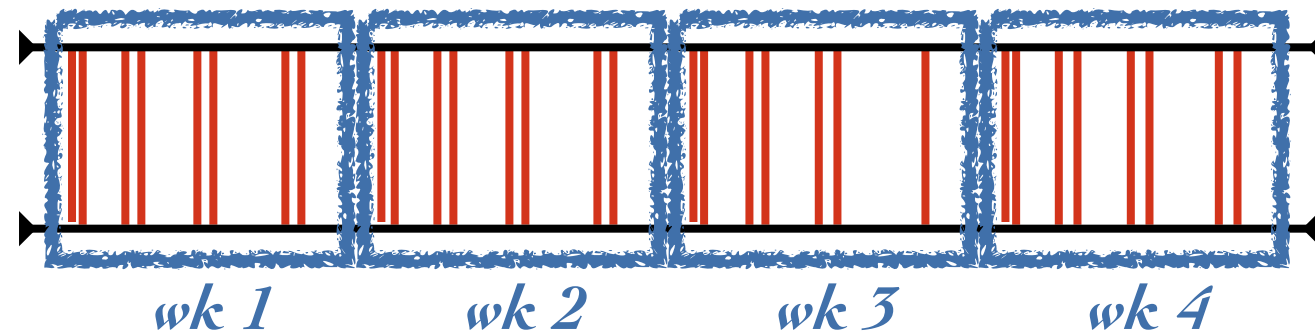
Event-based visit chronologies



- Many systems record visit data as **zero-duration events**
 - e.g., Foursquare checkins, transactions at retail stores, travel payment card swipes
- The data are also **sparse**; an individual rarely visits the same location more than **six or seven times a week**
- We need an **efficient** measure that handles **event-based** visit data that may be **sparse**

Quantifying regularity

...using IEl-irregularity



- **IEI-Irregularity:** “inter-visit interval irregularity”
- Approach adapted from **neural coding**
- Compare the **inter-event intervals** at the same **time of week**
- If the **inter-event intervals** are **similar** in each week, then the user’s visits to the location are considered **regular**

IEI-irregularity scores

score = 0.040

score = 0...

- perfect regularity
- the user visits the location the **same time each week**

score = 0.392

scores > 0...

- higher scores mean **more irregularity** in the user's visiting patterns

Results



Foursquare



Dartmouth
College

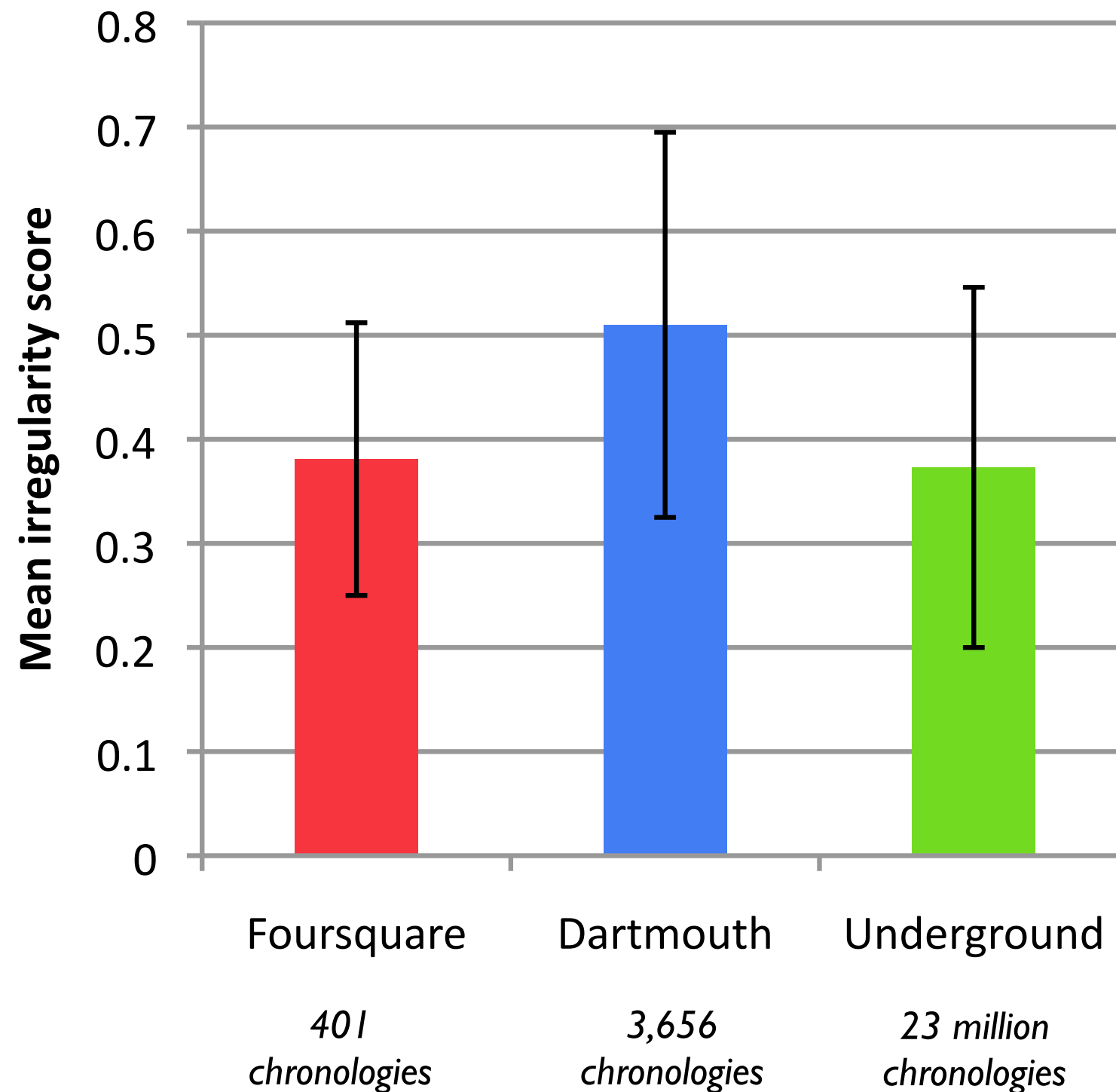


London
Underground

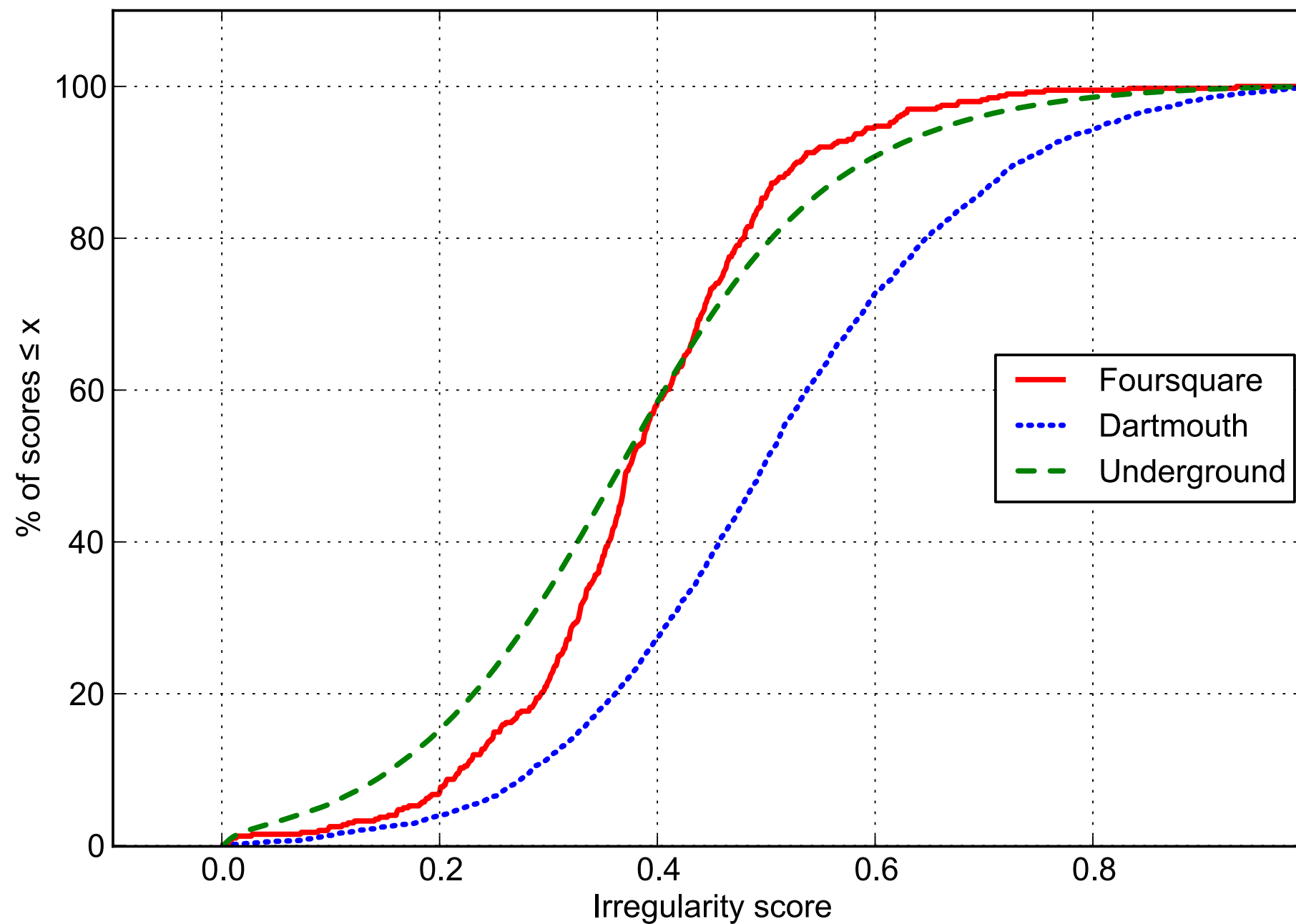
Scale	Visit type	Num. users	Num. locs.	Num. visits	Num. chronologies	Avg visits per chronology
Urban	Check in	293	336	4,640	401	11.6
Campus	WLAN access point association	1,681	391	229,300	3,656	62.7
Metrop.	Card swipe	1,167,363	270	58 million	2.3 million	26.1

- Only chronologies with at least **two visits per week** are considered
- All datasets represent 28-day periods

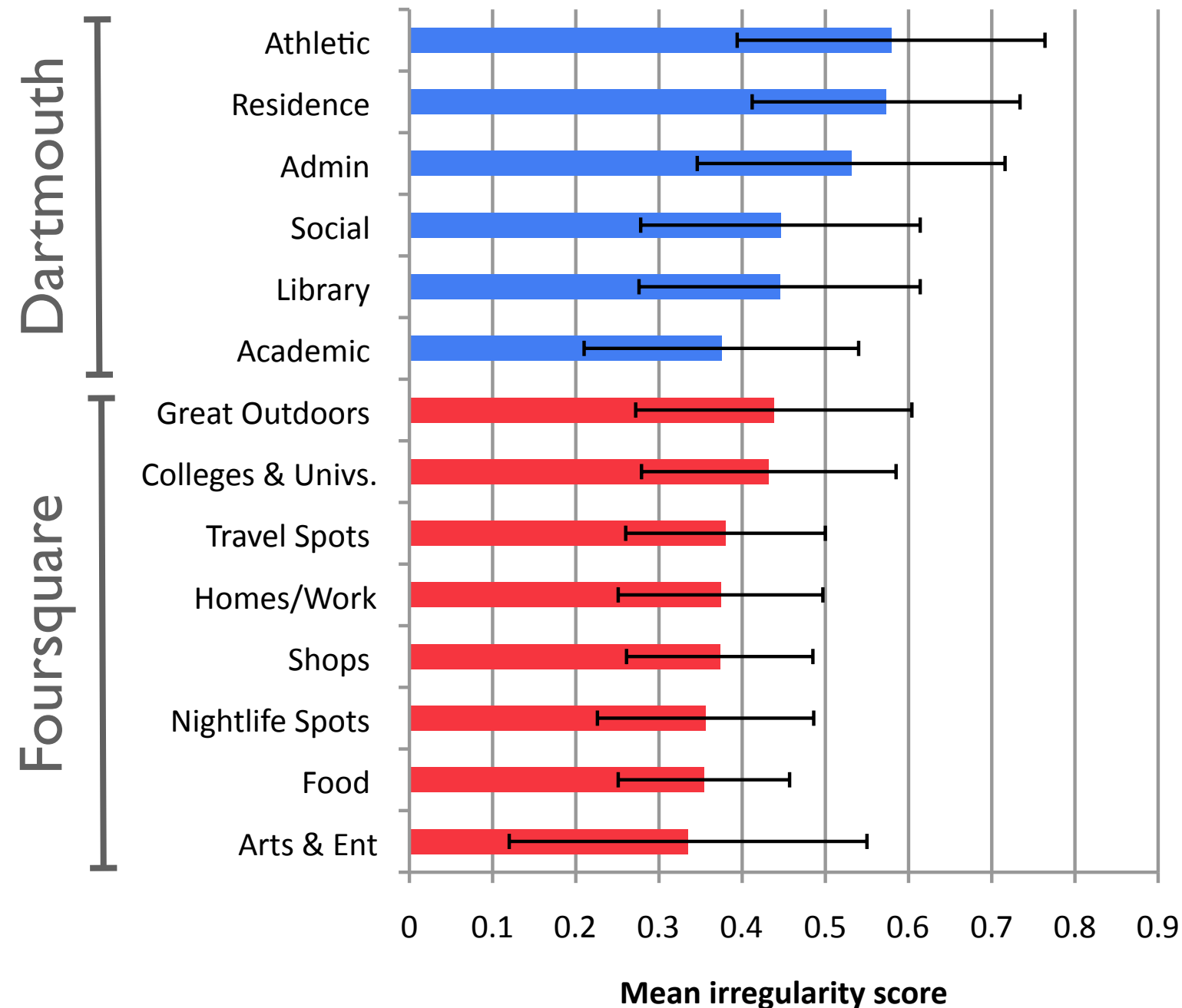
Dataset comparison



Dataset comparison



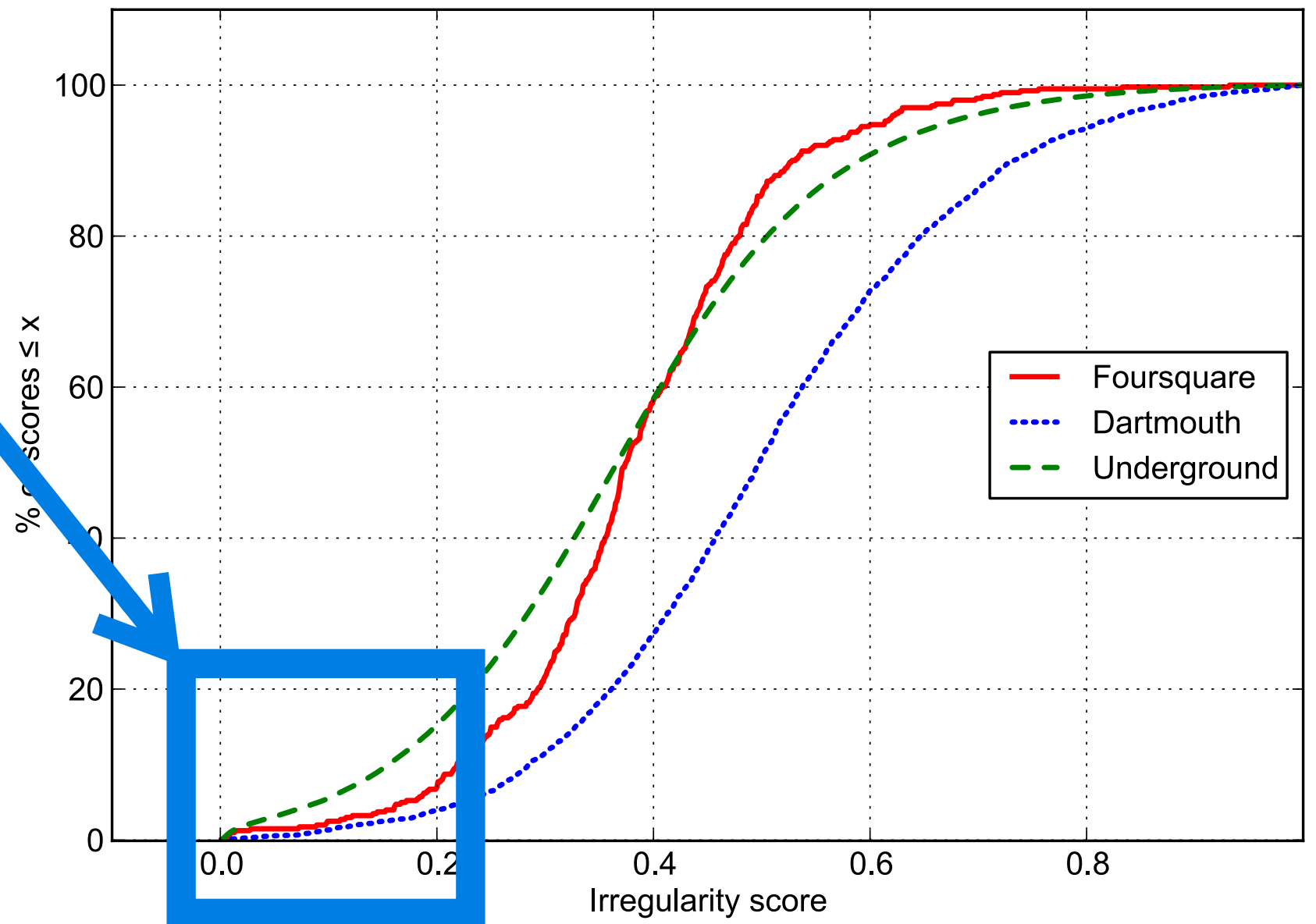
Comparison by location type



Very regular chronologies

- Number of 'very regular' chronologies (those with irregularity ≤ 0.2):

- Foursquare: 8.2%
- Dartmouth: 4.4%
- Underground: 17.4%



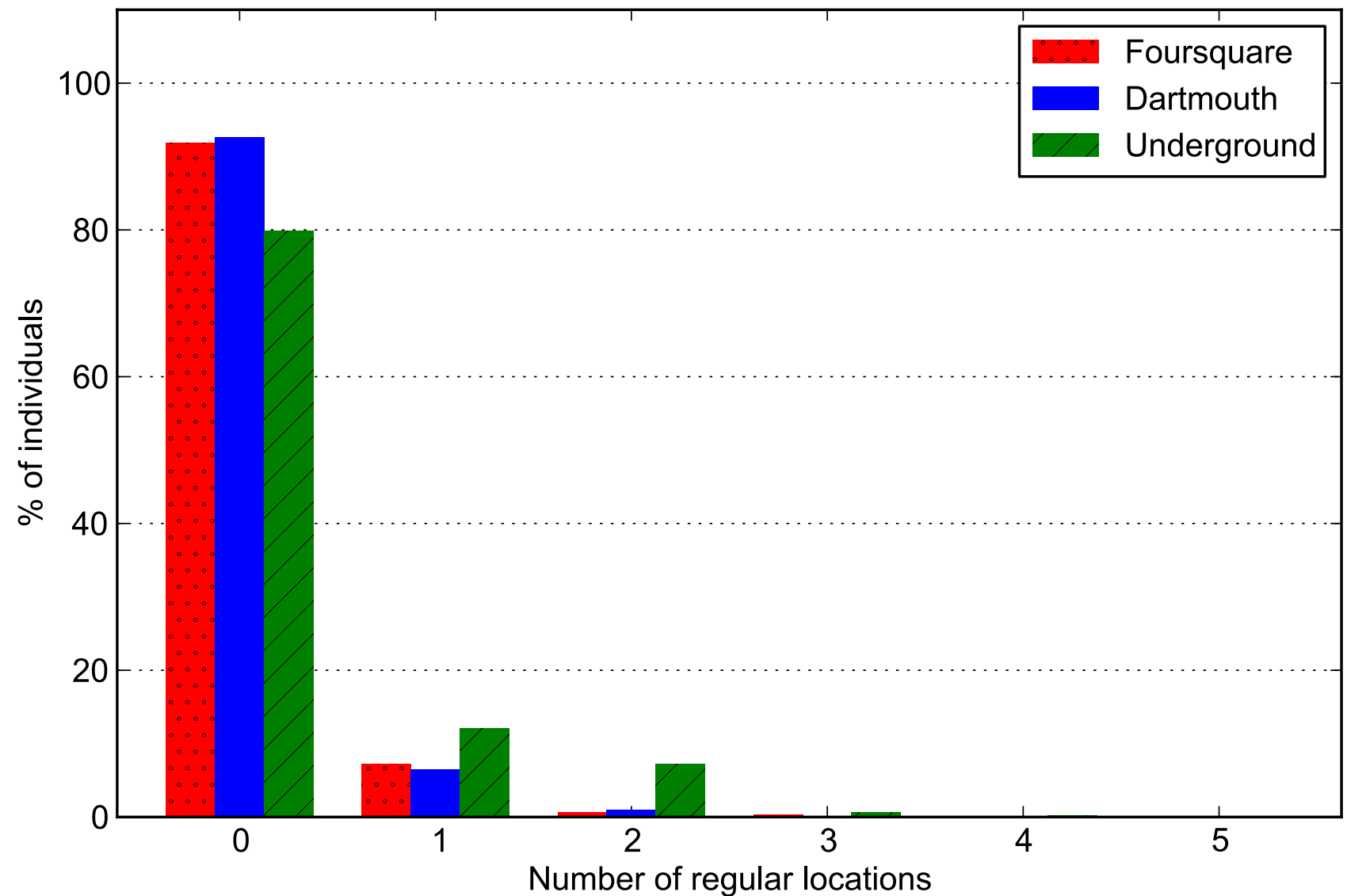
Very regular locations per user

- Number of users with at least one 'very regular' location:

- Foursquare
9.3%

- Dartmouth
8.2%

- Underground
21.2%

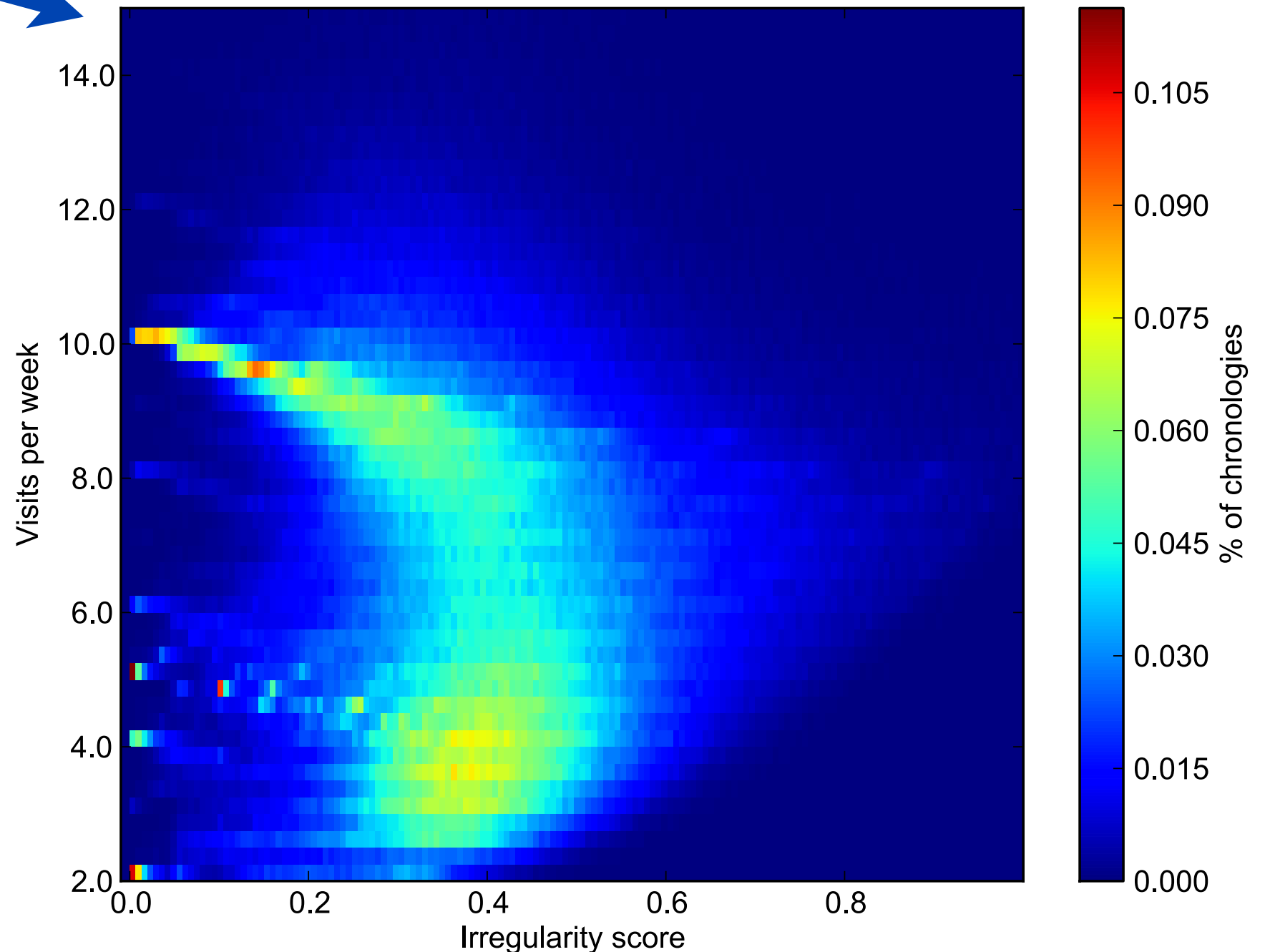


Frequency vs regularity

London Underground



- If you visit somewhere often, do you have a regular pattern with it?
- Self-reported surveys show that Underground passengers do not associate regular with frequent – what do the data say?

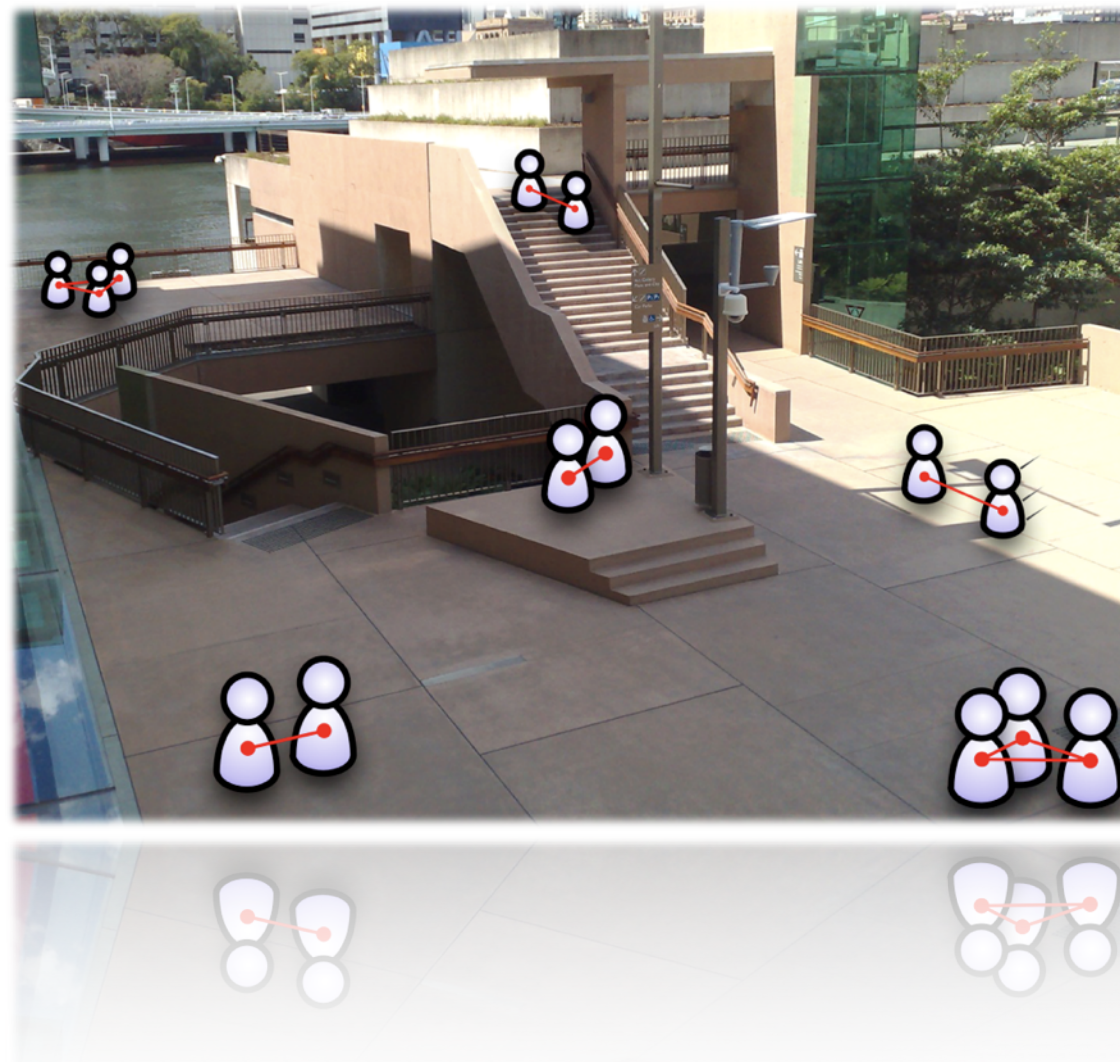


Visit patterns – summary

- IEl-irregularity: efficient measure for computing week-on-week irregularity in **event-based visit data**
- **Small core of users (8% to 21%)** in each dataset with **at least one regular location**
 - Core largest for an urban transit system
- Frequency and regularity have no linear correlation
- University campus access point visiting patterns least regular
 - Flexible and spontaneous student behaviour, and finer-grained movements
- Urban transit system most regular
 - Significant **commuter population** following rigid routines

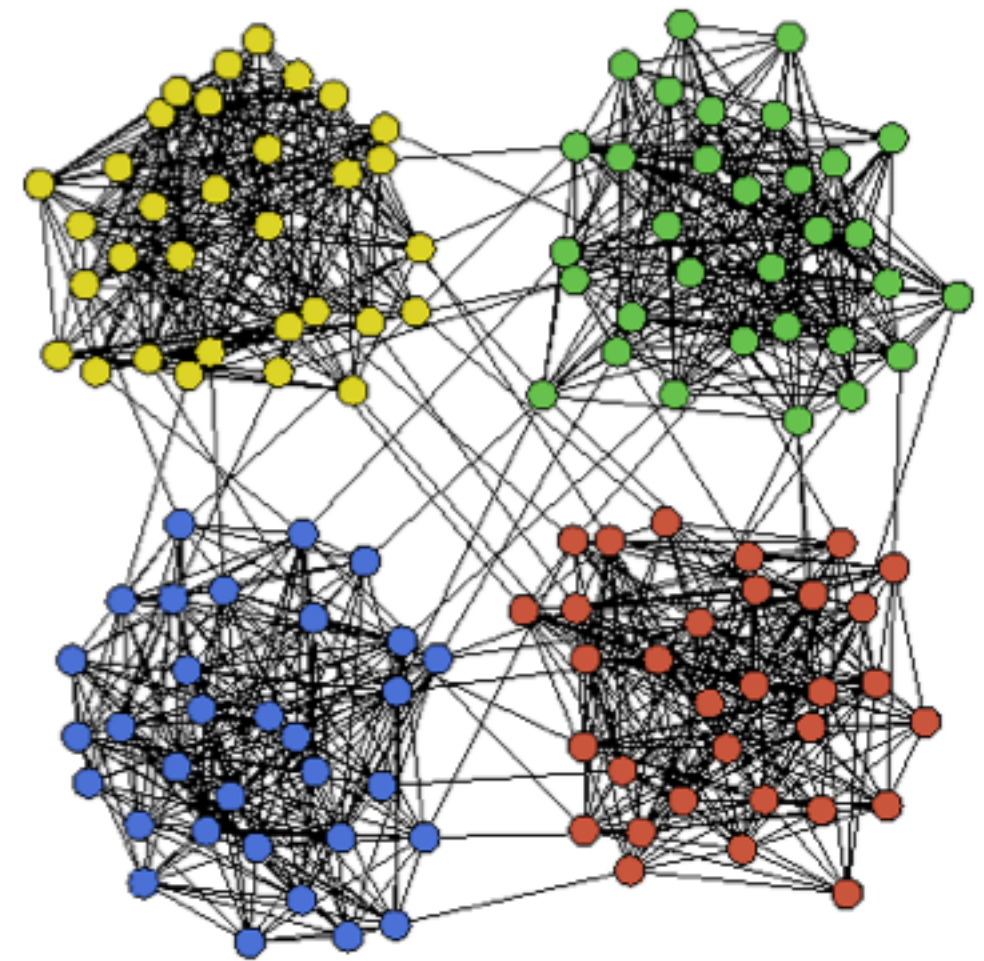
Part 2

Extraction of periodic encounter communities and evaluation of their content-sharing performance



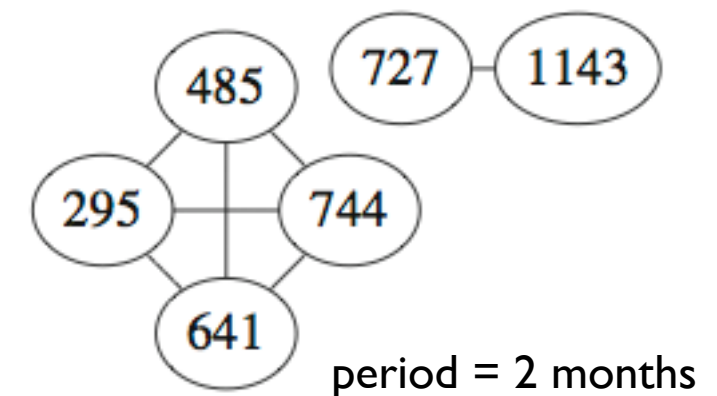
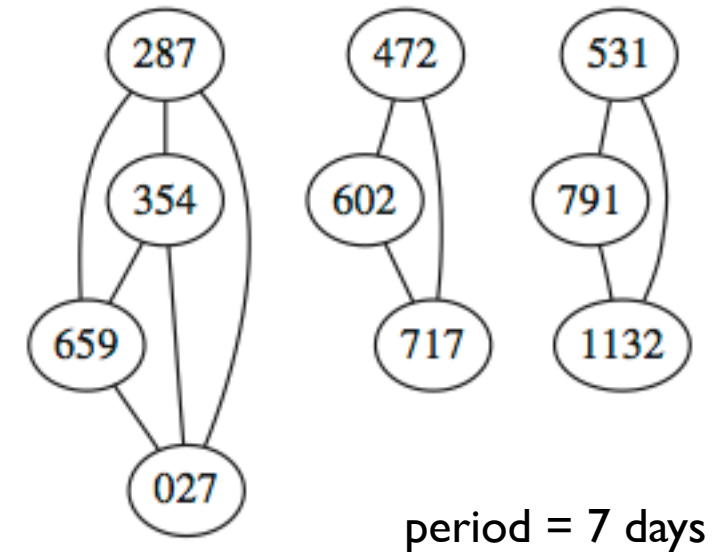
Perspective: static community detection

- Identify components in large graphs
- Global-knowledge, offline algorithms
- Static: single, time-agnostic graph
- Distributed algorithm used in oppnets content sharing



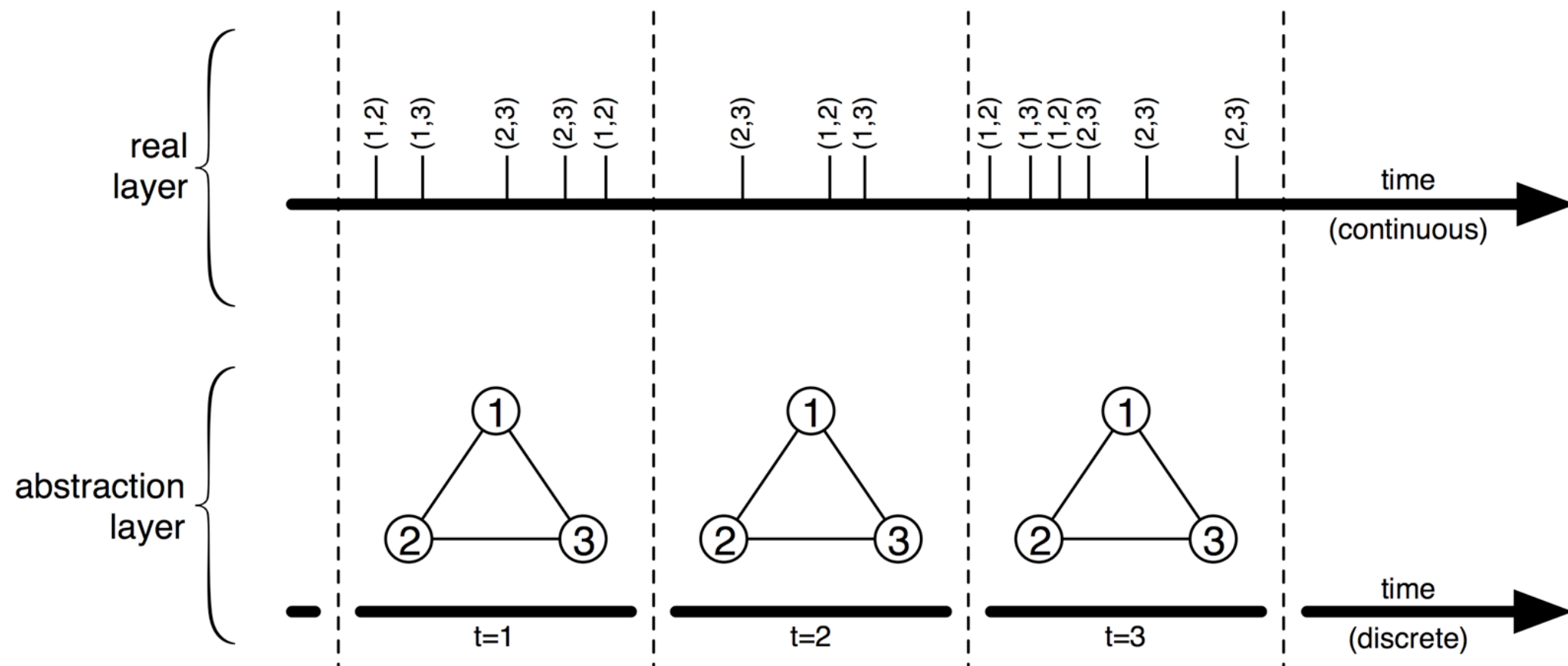
Periodic communities

- It is intuitive that the underlying behaviour of nodes results in **communities** of nodes **re-appearing regularly** in time
- Also evidenced in empirical datasets by **PSE-Miner** and other analyses
- We seek to join the concepts of **node communities** and **periodicity**
- **Decentralised** approach necessary in oppnets
- With automatic detection of **periods**



Dynamic encounter representation

- A **dynamic encounter network** is a time series of graphs
- Each graph is a **snapshot of encounters** occurring during a time interval



Periodic encounter community

- We formalise a Periodic Encounter Community (PEC) as

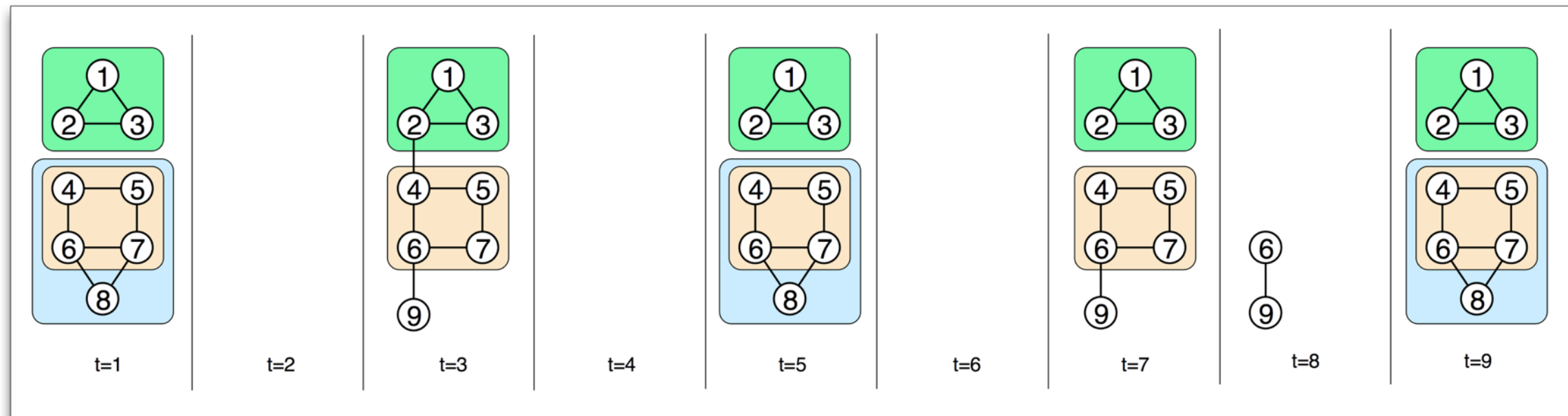
$$\langle C, S \rangle$$

- where
 - C is a **connected** graph (the **community**)
 - S is the **harmonic information**

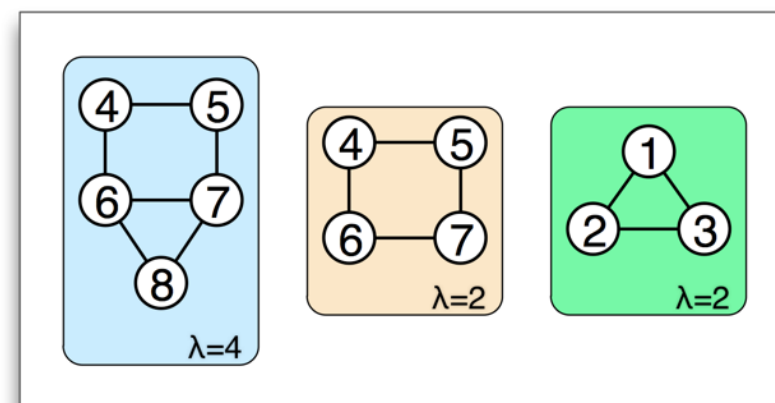
$$S = (t_{start}, t_{end}, \lambda)$$

PEC example

Example Dynamic Network

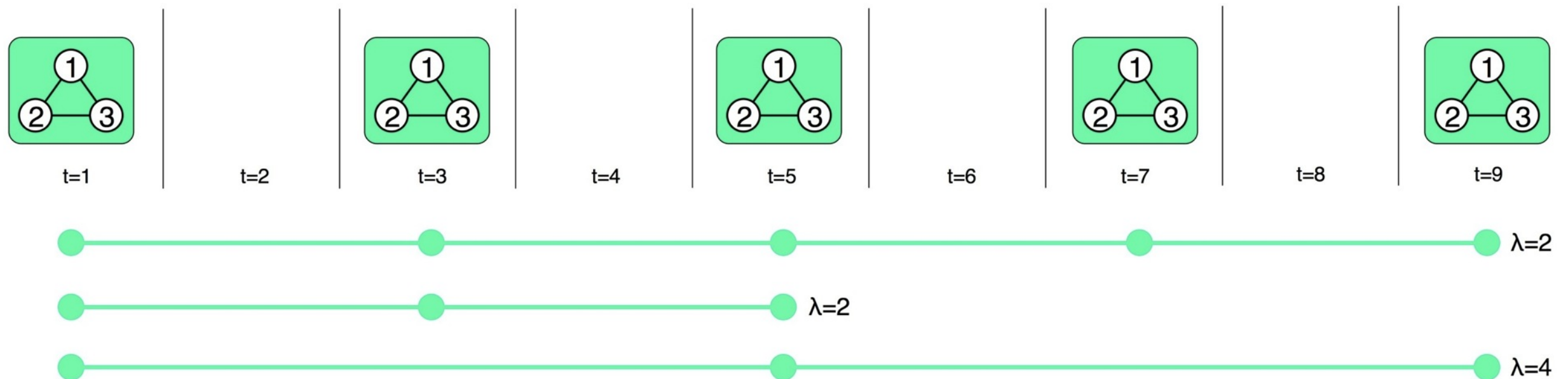


Periodic Encounter Communities



PEC redundancy

- Some PECs capture more information than others
- One PEC may **subsume** another's information



- **Harmonic maximality:**
 - Multiple ways to fit harmonic information to the same community, but only one is **parsimonious**

Maximality and parsimony

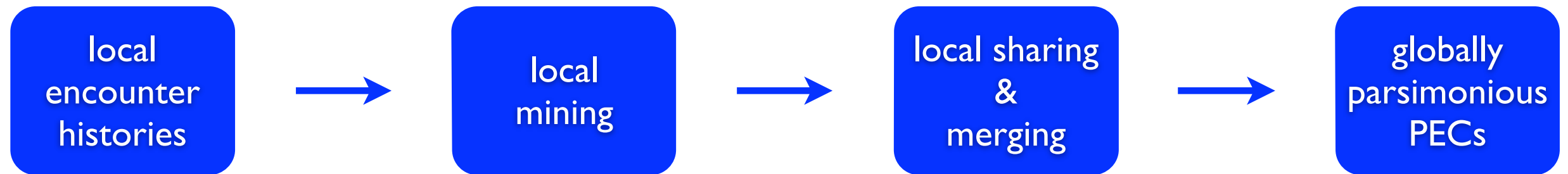
- **Harmonic maximality:**
 - Community does not exist for factors of the period, nor can it be extended in time
- **Structural maximality:**
 - Cannot add edges or nodes to the community and still maintain its existence in the dynamic network
- **Parsimony:**
 - A PEC is **parsimonious** if it is both **harmonically** maximal and **structurally** maximal

Decentralised PEC-D problem

- Decentralised PEC Detection is the problem of having all nodes detect the **parsimonious PECs** they belong to, without **global knowledge** of the network

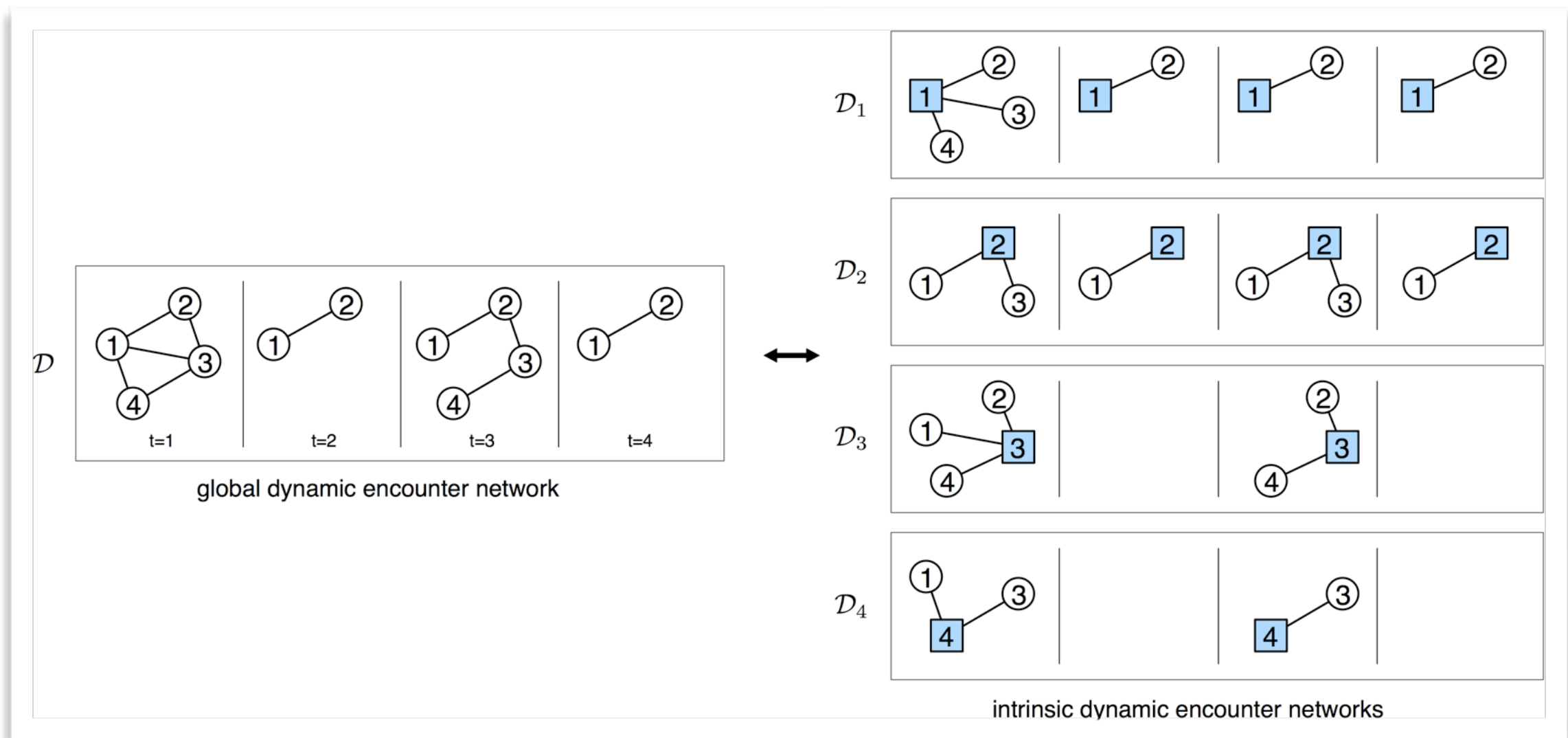
Decentralised PEC detection algorithm

Algorithm Overview



- Local Mining:
 - Obtain PECs that are parsimonious in their **local encounter histories**
- Local Sharing:
 - Nodes share and combine their intermediate parsimonious PECs when they meet
- Over time, nodes build towards the PECs that are parsimonious in the **global dynamic graph**

Intrinsic Dynamic Networks



- Global dynamic network can be decomposed into **intrinsic dynamic networks**
- Intrinsic DN corresponds to the encounter information **directly observable** by a node

Local Miner Algorithm

- Invertible map from graphs to sets of integers
 - Edges and nodes given unique integer identifiers
- Becomes a problem of mining **periodic subsets** in a time series of integer sets
 - **Periodic pattern mining in temporal data mining field**
- Polynomial time complexity
- Local returns **locally-parsimonious PECs**

Joining PECs

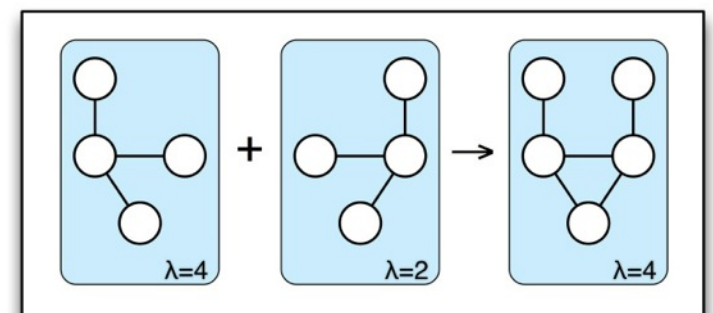
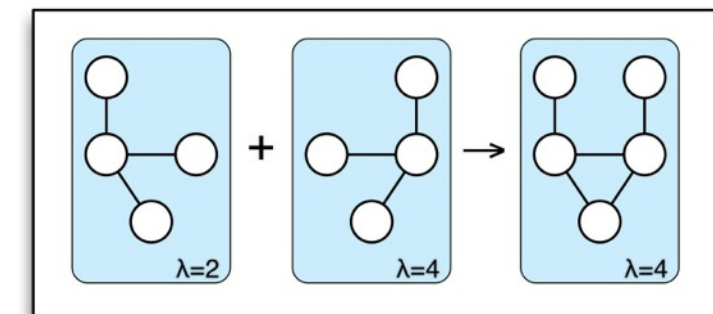
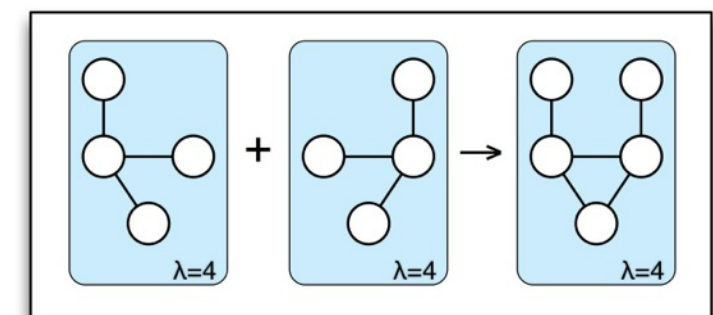
- Two PECs are **compatible** if the following hold:
 - their communities intersect
 - the PECs are harmonically equal, or one harmonically subsumes the other
- If compatible, there are three generation cases:

case		action
harmonic equality	merge communities	keep harmonic information
P1 harmonically subsumes P2	merge communities	harmonic information from P2
P2 harmonically subsumes P1	merge communities	harmonic information from P1

Opportunistic Construction

- Each node holds local Knowledge Base (KB) of its PECs so far
- Node only holds non-subsumed PECs which node itself belongs to
- On encounter, a pair of nodes:
 - Share KBs
 - Generate candidate PECs
 - Store any more-maximal candidates
 - Remove any redundancies

PEC Generation Cases



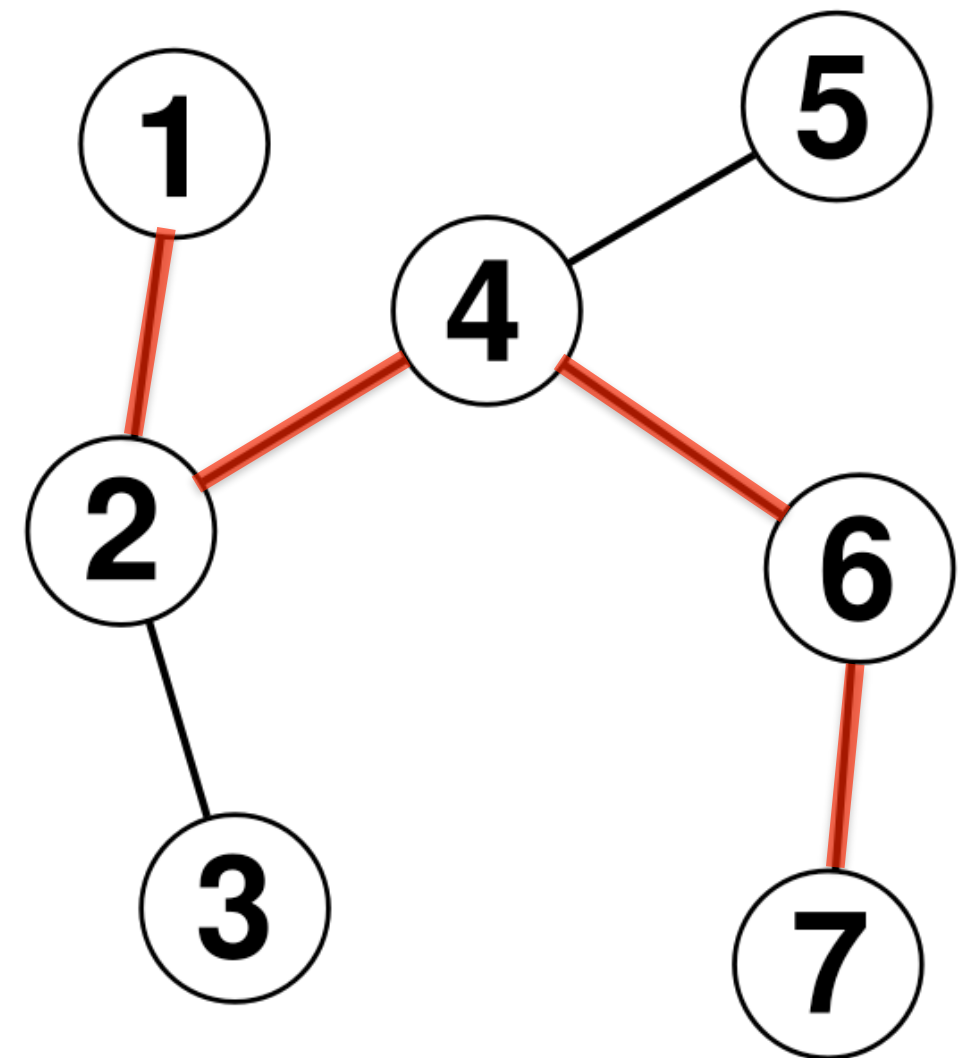
Analysis of PECs

Questions

- How prevalent are periodic encounter patterns in human networks?
- How does the presence of periodic encounters affect information flow?
- Can periodic patterns be detected and used to improve content sharing in opportunistic networks?

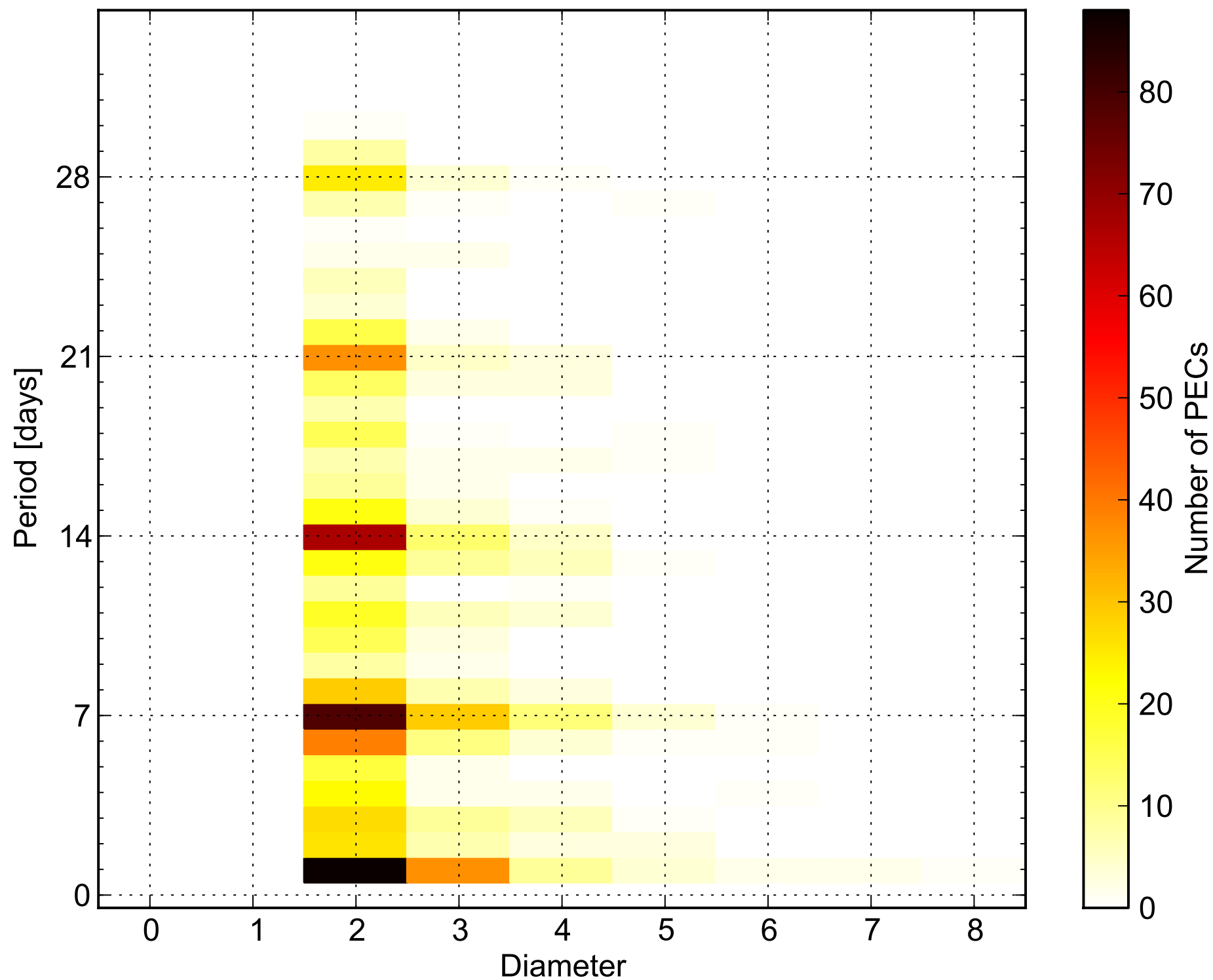
Properties of PECs

- **Period:** the gap between reappearances of the community (24 hrs, 7 days, etc.)
- **Diameter:** the distance between the most distant nodes in the community

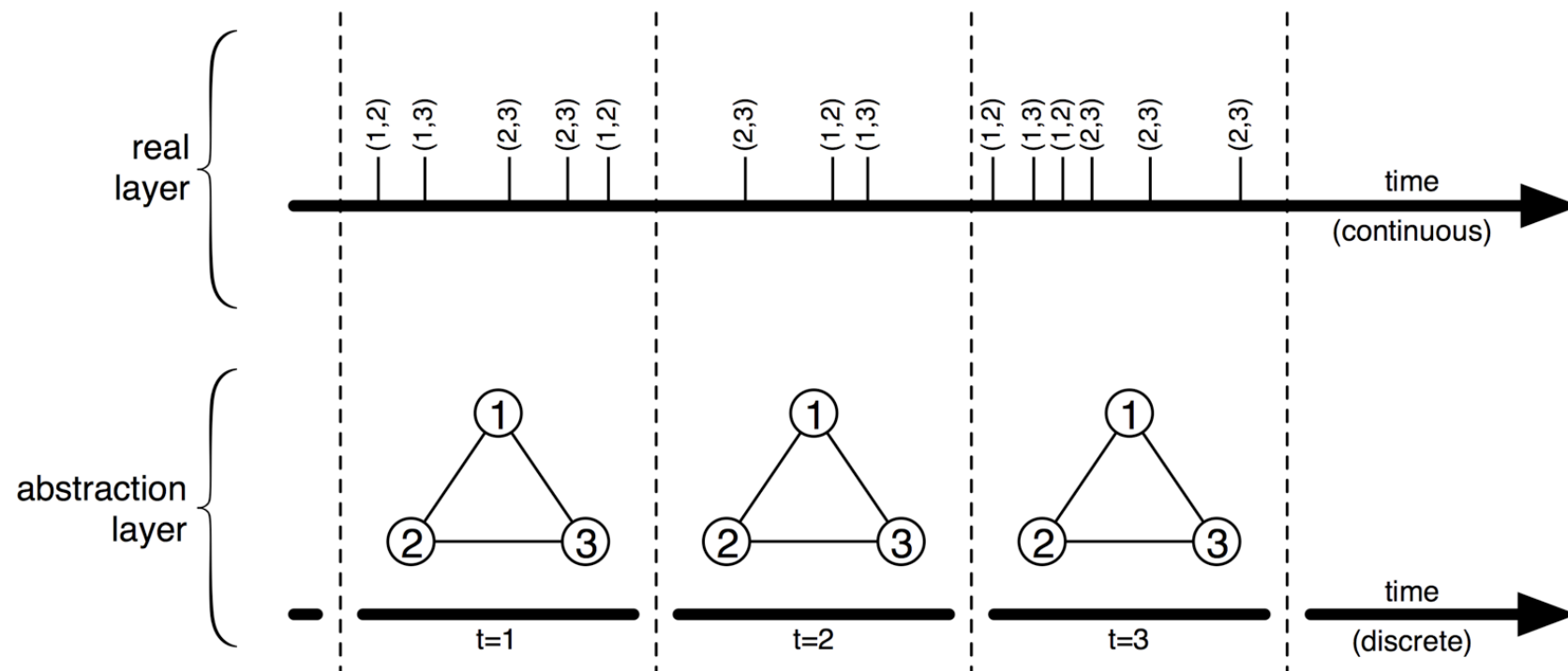


diameter = 4

Heatmap titled "Joint frequency distribution of PEC diameters and periods". The x-axis is labeled "Diameter" and ranges from 0 to 8. The y-axis is labeled "Period [days]" and ranges from 0 to 28. A color bar on the right indicates the "Number of PECs", ranging from 0 (light yellow) to 80 (dark red/black). The highest frequency (darkest color, ~80) is concentrated in the region where diameter is between 1.5 and 2.5 and period is between 7 and 14 days. Other notable peaks occur at lower diameters and higher periods, and at higher diameters and lower periods.



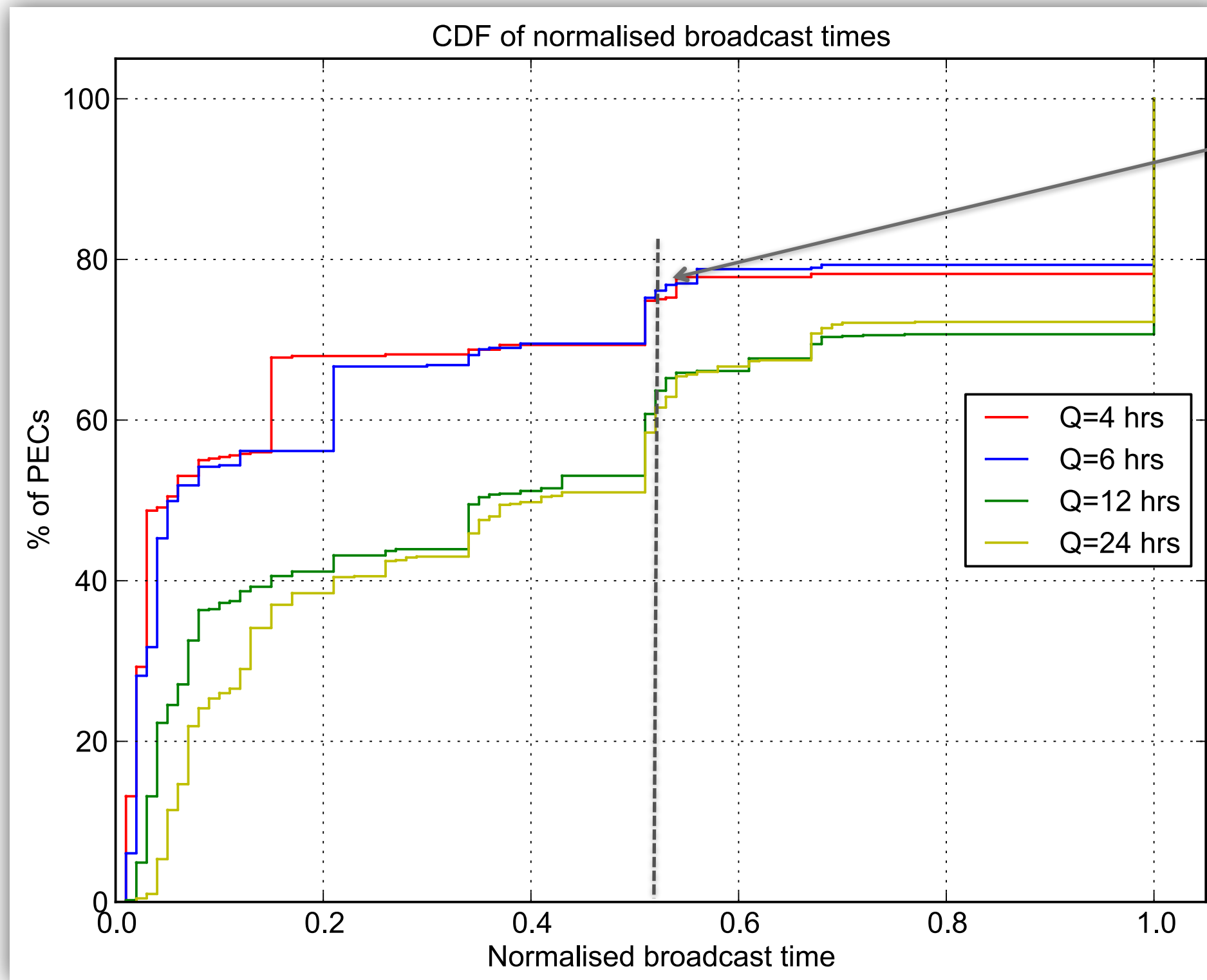
Broadcast Time



- **‘Token broadcast’** – a tool to measure **construction time** and **information sharing capacity** of PECs
- Broadcast relies on encounters between nodes
- The underlying ordering of edges influences how tokens propagate to nodes

Token broadcast

- **Diameter and period** give us a **theoretical worst-case** for the time needed for all nodes to send tokens to each other
- **In practice**, how does token broadcast compare to the worst-case?



in 75% of PECs, broadcast took less than 1/2 the worst-case time

Other observations

- Running experiments with a larger granularity (snapshot size) leads to slower broadcast
- Although PECs with larger diameter have a larger worst-case broadcast time, they are less likely to reach it

PECs – summary

- Globally parsimonious PECs can be mined decentrally, and with automatic periodicity identification
- Time for globally parsimonious PEC construction is bounded by PEC period and diameter
- On real data (Bluetooth encounters at MIT), construction time is much better than the analytic worst-case

Limitations

- The PEC data mining approach enables automatic period detection

but...

the discrete-time representation results in **loss of temporal resolution** and is **sensitive to noise**

Regular encounter communities (RECs)

*Using IEI analysis as an alternative
to a discrete-time data mining approach*

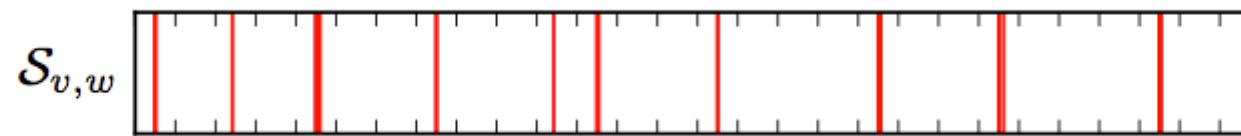
REC

- IEI analysis allows us to extract patterns in a time-resolved manner
- Let's replace the discrete-time harmonic information used in PEC detection with a time-resolved measure based on inter-event intervals (IEIs)
- Assumption: we're looking for encounter patterns at a **single period** that we select a **priori**

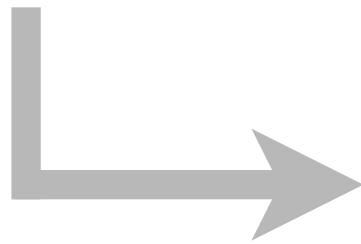
Regularity mask

- We build a **regularity mask** which represents the times-of-week where a **whole community** is regular
- We start by constructing a regularity mask for a pair of nodes...

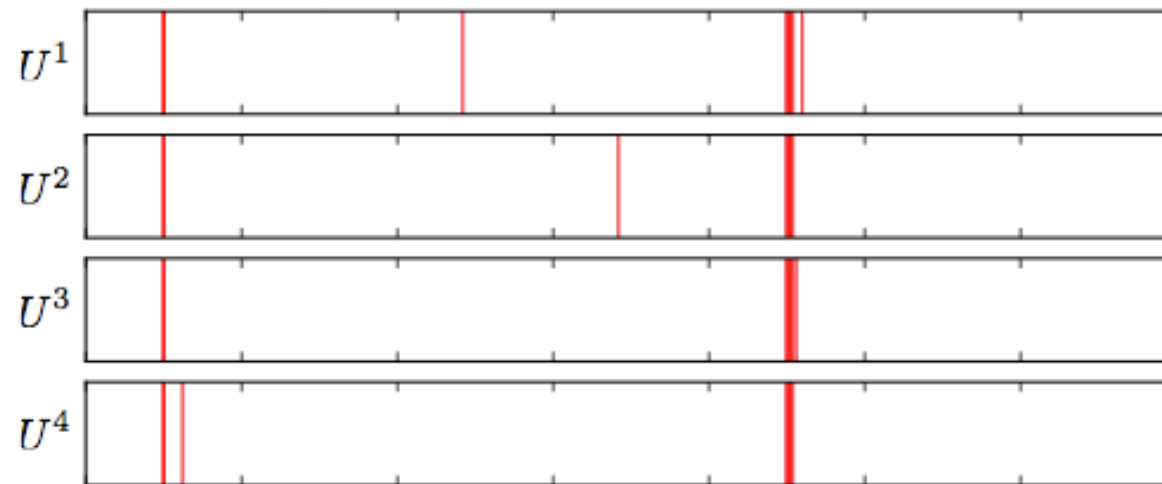
Pairwise regularity mask construction



(a) An example chronology $\mathcal{S}_{v,w}$ depicting encounters between individuals v and w over 28 days.

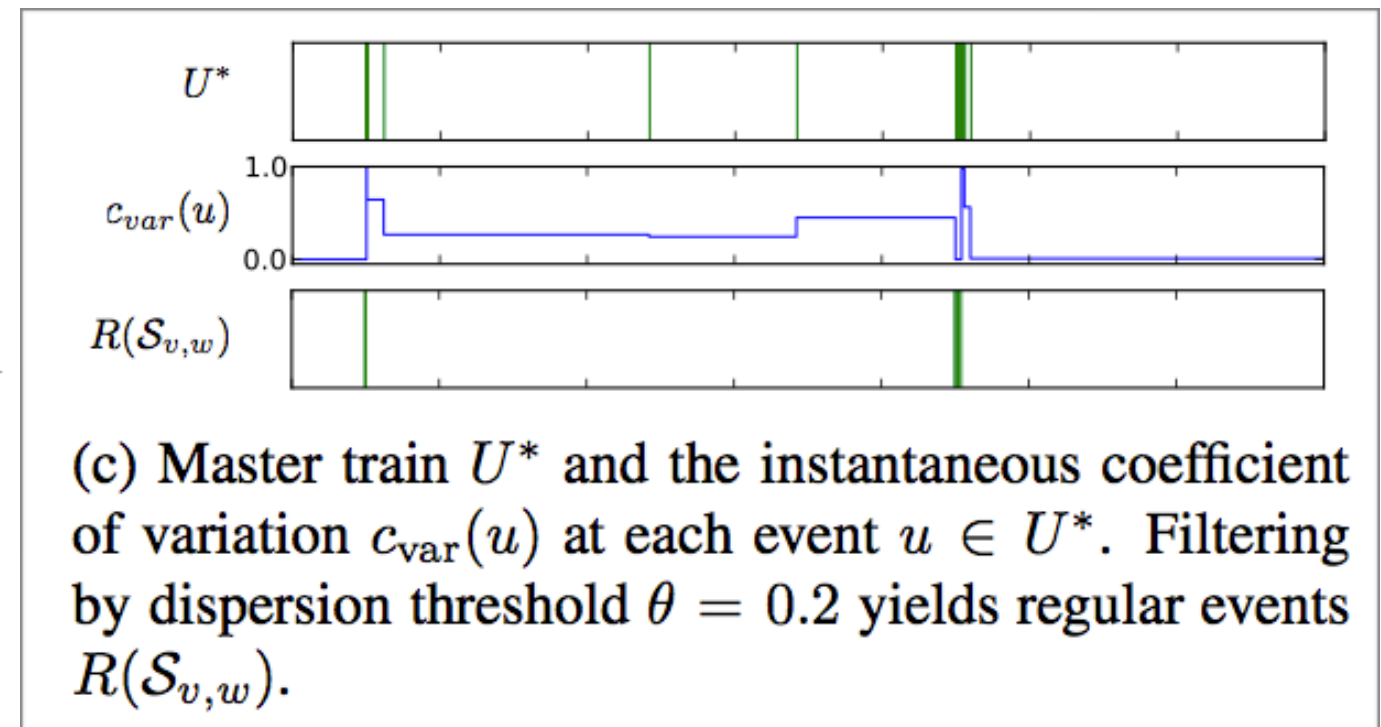
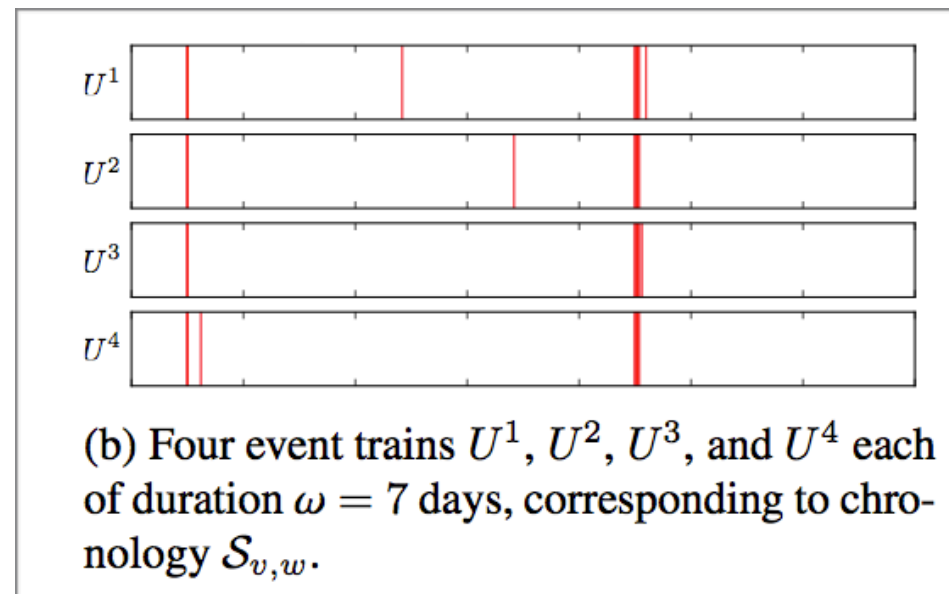


- Step 1:
segment **encounter
chronology** into
windows
- period = 7 days
(i.e., compare week by
week)



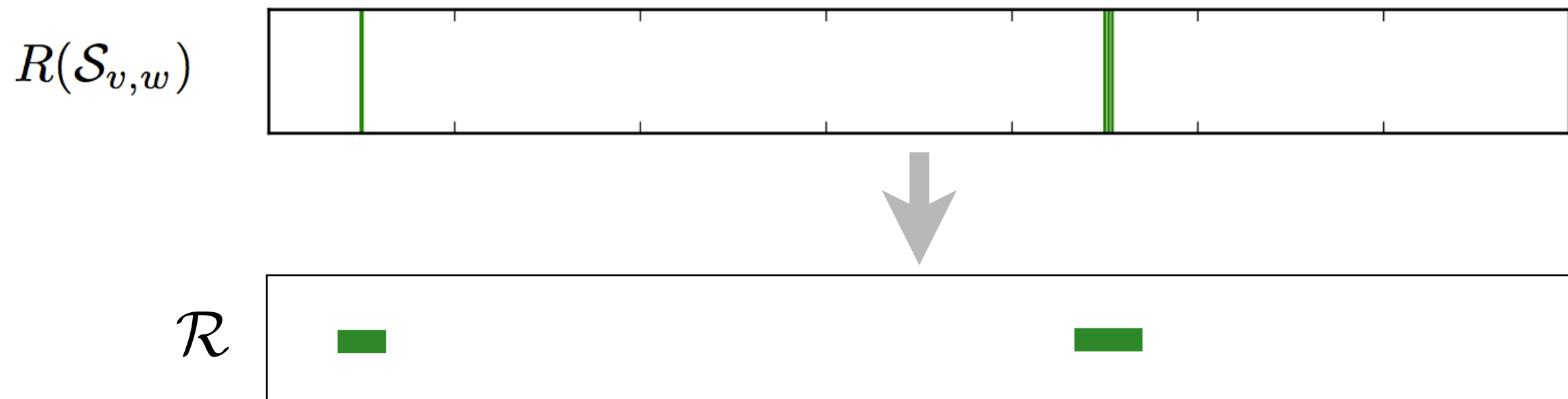
(b) Four event trains U^1 , U^2 , U^3 , and U^4 each of duration $\omega = 7$ days, corresponding to chronology $\mathcal{S}_{v,w}$.

Pairwise regularity mask construction



- Step 2:
- Based on the **time-of-week dispersion** in IEI (inter-event interval) values, we can determine which encounter events are part of **consistent behaviour** and which are not
- Result = subset of encounters between these two individuals that we regard as **regular**

Pairwise regularity mask construction

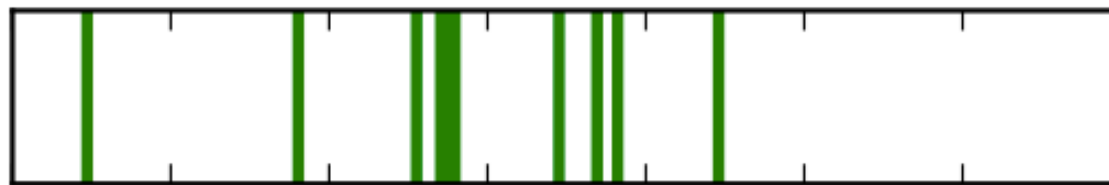
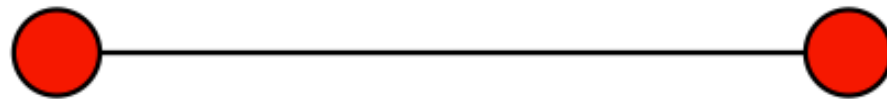


- Step 3:
- Inflate the regular events so that we can perform **regularity mask intersection** when constructing multi-node communities

REC – definition

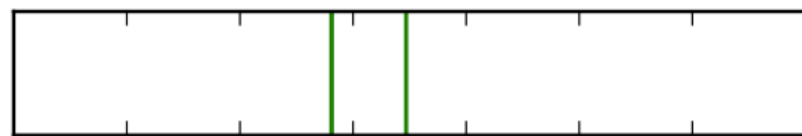
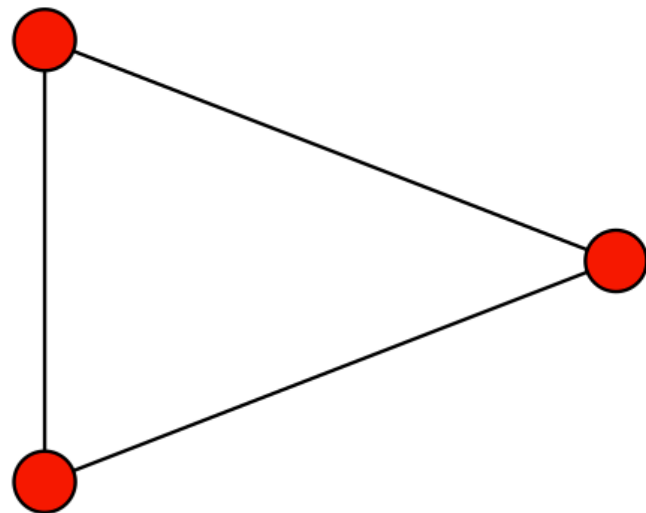
- A regular encounter community (REC) is a community whose intersection of regularity is masks is non-empty
- In other words, the individuals in the community all share a weekly encounter pattern
- We can re-use a lot of the decentralised PEC detection algorithm, replacing a few components:
 - Structure = community (a connected graph) (same as before)
 - Harmonic information = regularity mask
 - REC combination = by **graph union** and **regularity mask intersection**

REC examples

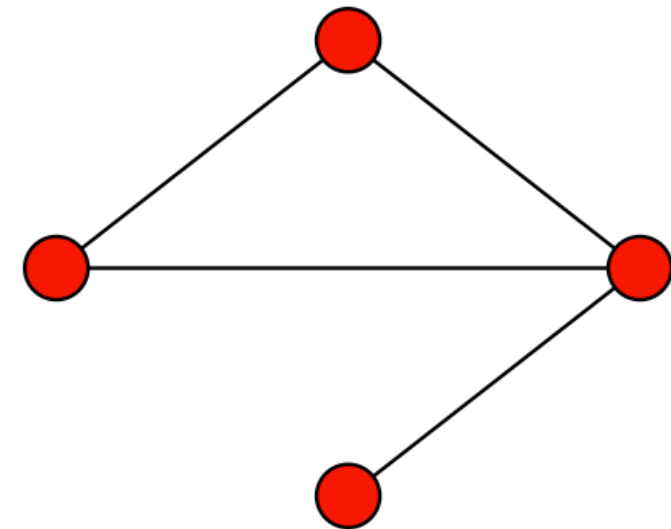


$$\mathcal{R}_1 = \langle C_1, R_1 \rangle$$

REC examples



$$\mathcal{R}_4 = \langle C_4, R_4 \rangle$$



$$\mathcal{R}_5 = \langle C_5, R_5 \rangle$$

RECs in the Reality Mining dataset

- 210 RECs detected for the chosen four weeks
- 76% of participants belonged to at least one REC
- 64% of RECs contained two to three individuals
- Diameters typically small, but larger than PECs

Density of communities

- How close to being a clique is a REC?

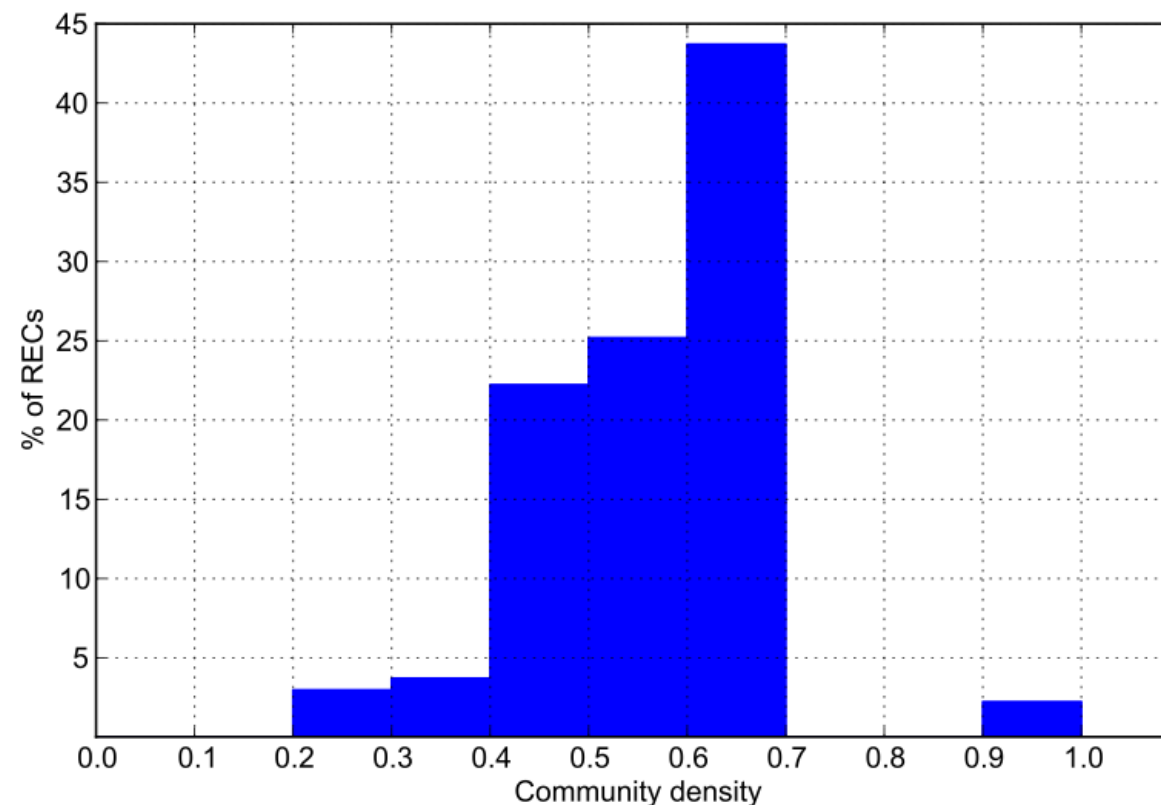


Figure 6.5: Distribution of community density for REALITY RECs consisting of at least three nodes. For a REC with community $C = (V, E)$, community density is given by $\frac{2|E|}{|V|^2 - |V|}$.

What time-of-week are individuals most regular?

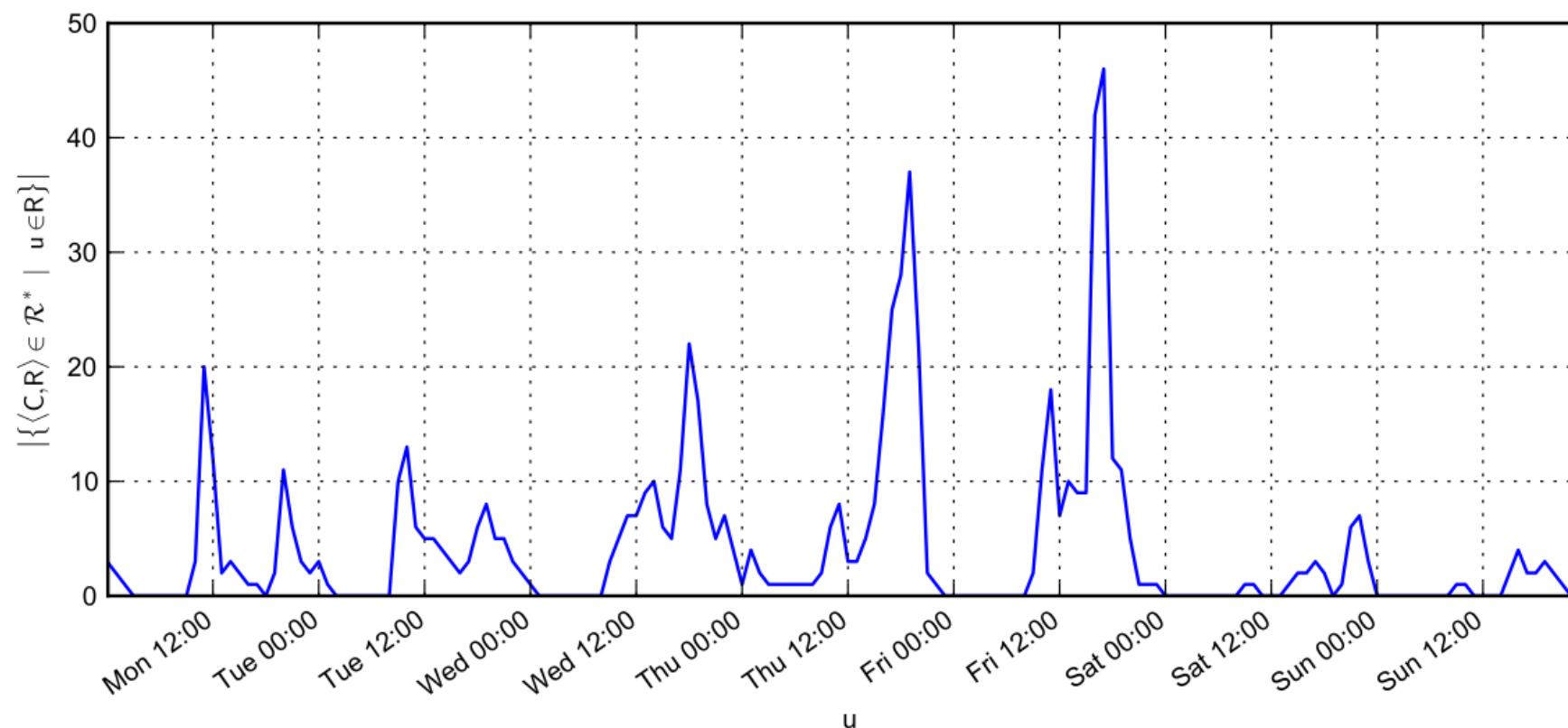


Figure 6.7: Distribution of regularity masks belonging to REALITY RECs throughout the week. Letting $\mathcal{R}^*$ denote the set of all RECs, the number of regularity masks that include the time of week u is given by $|\{\langle C, R \rangle \in \mathcal{R}^* \mid u \in R\}|$.

Correspondence between RECs and PECs?

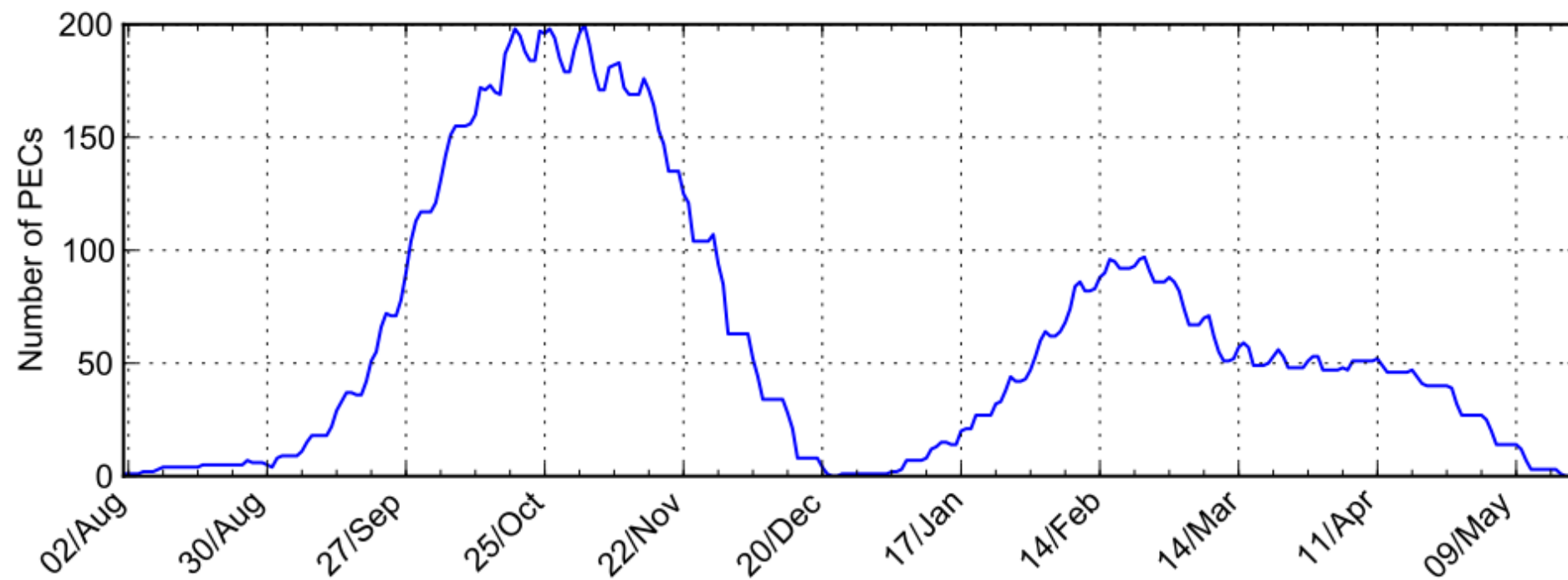


Figure 6.2: Occurrences of periodic encounter communities (PECs) found in the REALITY dataset that have a period of seven days. The height of the curve at date t corresponds to $|\{ \langle C, S_{(i,p,n)} \rangle \in \mathcal{P}^* \mid i \leq t \leq i + (n-1)p \wedge p = 7 \text{ days} \}|$ where $\mathcal{P}^*$ is the set of all PECs extracted from REALITY with granularity $Q = 24$ hrs.

- Chose a four-week duration in the data
- Compare PECs with period of one-week to RECs with window of one week
- Do they find the same behaviour? Are RECs able to identify more patterns?

Correspondence between RECs and PECs?

- 58% of PECs also appeared as a REC
- 14% of RECs also appeared as a PEC

RECs for token broadcast

- PEC is a stricter in its periodic requirement:
the community **must** meet in each periodic timestep
- RECs have weaker requirement, and so token broadcast suffers
- Only 35% of RECs reached full broadcast after one week
- 32% RECs **failed** to reach full broadcast after the full (four week) duration
(cf. 0% for PECs)

RECs – Summary

- Re-used periodic encounter community (PEC) decentralised algorithm for REC constructions, with some changes:
 - regularity mask instead of periodic timesteps
 - new local miner algorithm
- RECs give us time-resolved periodic patterns (higher temporal resolution than PECs)
- RECs capture majority (but not all) of the PEC patterns, plus more
- Due to less-strict assertion on encounter timing, information sharing (token broadcast) is slower in RECs

Future work

- Visit patterns from the location perspective
- A CRUD-ified protocol for decentralised PEC detection
- Temporal infrastructure for oppnets

Thanks for listening!

Any questions?



**Various work in
collaboration with:**

Roger Whitaker
Stuart Allen
Martin Chorley
Walter Colombo

Matt Williams

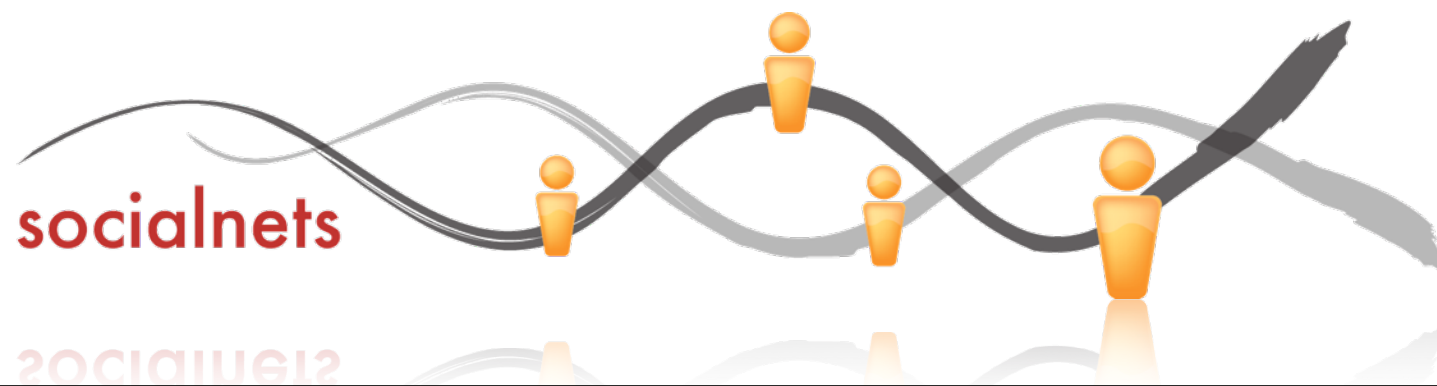
www.mattjw.net

M.J.Williams@cs.cardiff.uk

[@voxmjw](#)

www.gplus.to/mattjw

And supported by...



www.recognition-project.eu

www.social-nets.eu

Attribution

Foursquare maps:

<https://foursquare.com/>

User icons:

UX People stencil by "jcallender"

<http://graffletopia.com/stencils/639>

Students in class:

FOSDEM 2008 main lecture theatre

http://commons.wikimedia.org/wiki/File:FOSDEM_2008_Main_lecture_theatre.jpg

Crowd wearing masks:

http://www.ickypeople.com/2009_04_26_archive.html

Coffee shop counter:

"Counter stocked for opening day" by Buz Carter

<http://www.flickr.com/photos/pizzabytheslice/2320006035/in/photostream/>

Foursquare pub icon:

<https://foursquare.com/>

Foursquare logo:

<https://foursquare.com/about/logos>

Access point icon:

By IconShock

http://www.iconfinder.com/icondetails/45228/128/access_point_router_icon

London Underground logo:

<http://en.wikipedia.org/wiki/File:Underground.svg>