P-N-RMiner: A Generic Framework for Mining Interesting Structured Relational Patterns

Jefrey Lijffijt, Eirini Spyropoulou, Bo Kang, Tijl De Bie

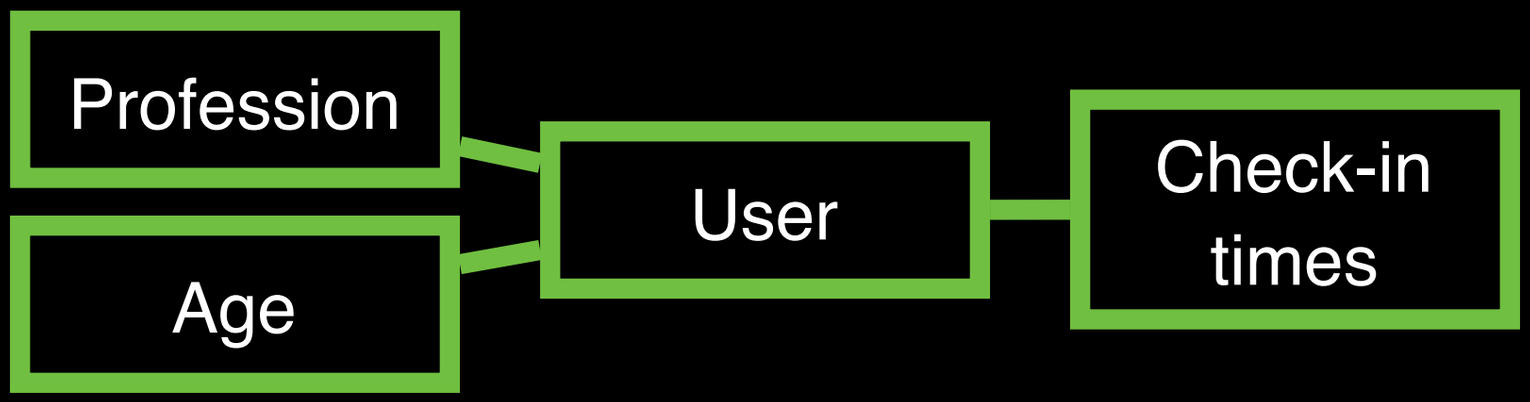
University of Bristol

[JL, BK, TDB → Ghent University, ES → Barclays]

Motivation

Example

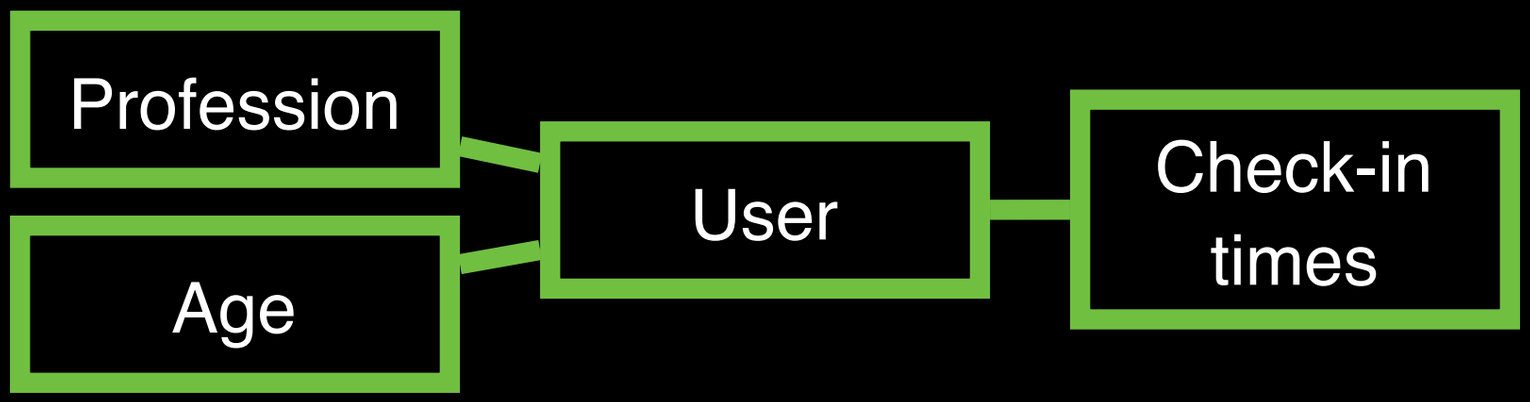
Suppose we have some data,



and we are interested in lifestyle patterns.

Example

What will we find?

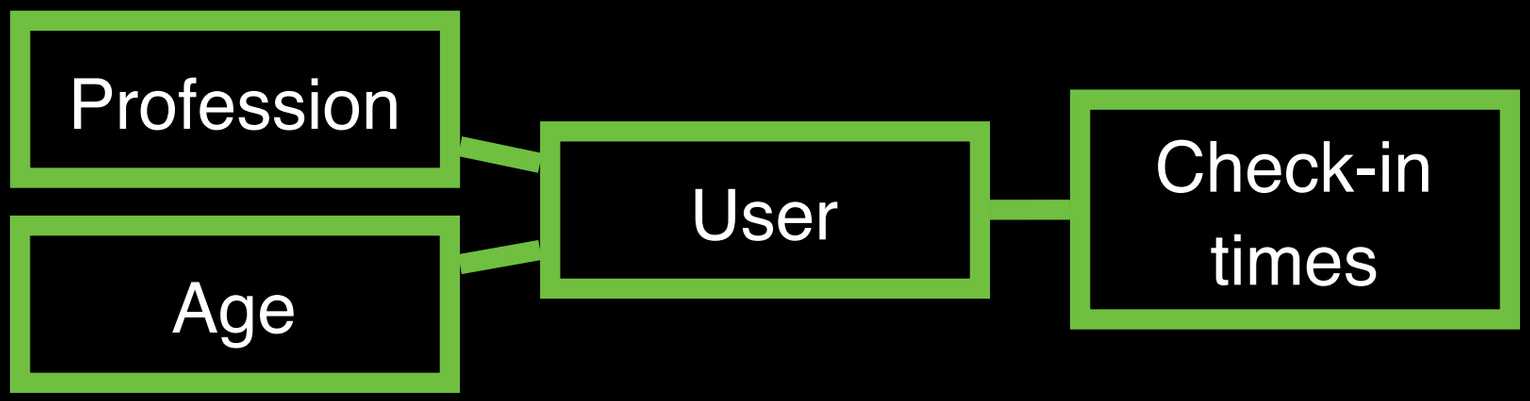


For example: “many people aged 30–45 check in somewhere both between 7.30 and 8.30 in the morning and between 11.30 and 12.30 around noon”

Could be interesting if unexpected

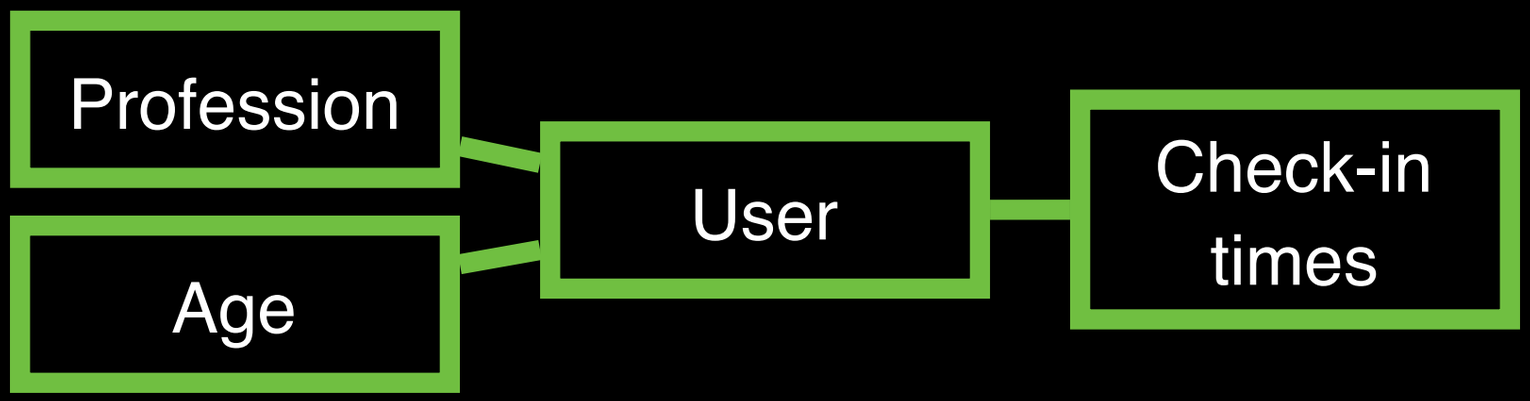
Example

Complicated data



Example

Complicated data



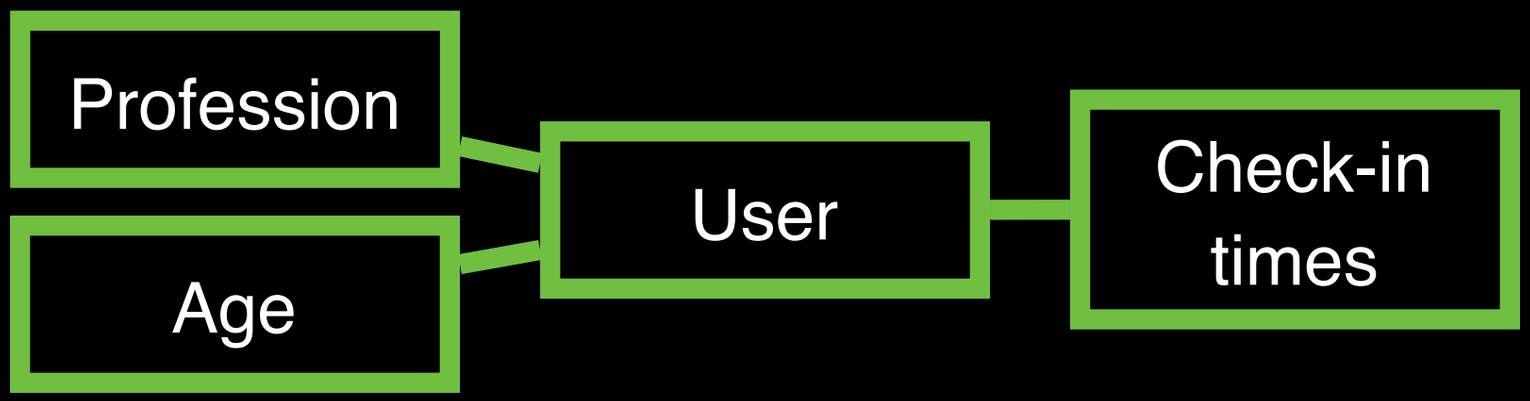
It is relational

Multiple professions, many check-in times

Cannot be flattened without loss of information

Example

Complicated data



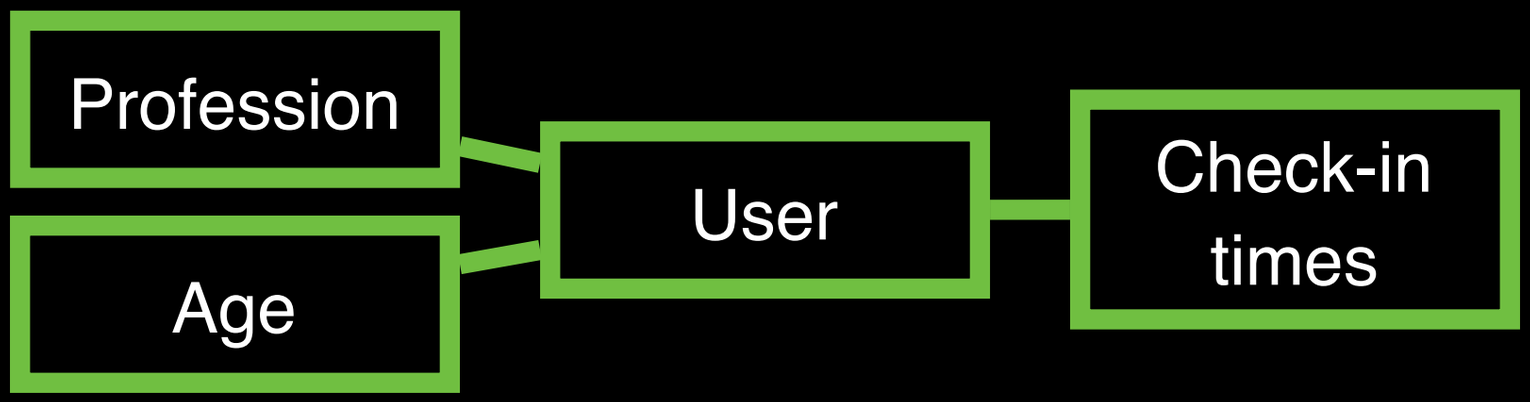
It has structured attributes

Time of day, age, professions (hierarchy or DAG)

No pattern mining framework deals with all these

Example

Complicated data



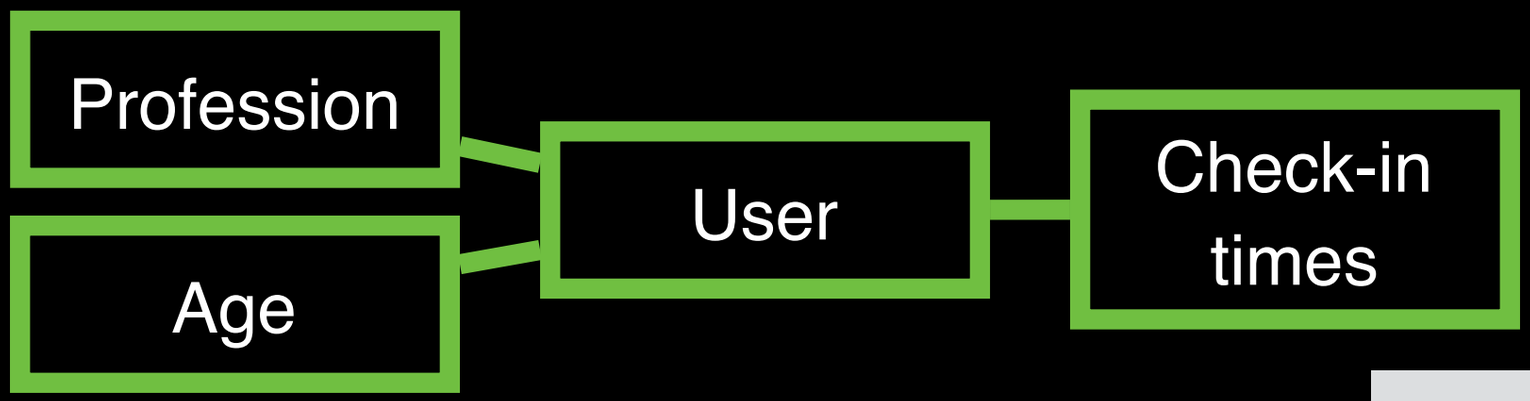
Typical solution: choose a granularity, discretise

Different patterns may need different granularity

Worse, some patterns require *mixed granularity*

Example

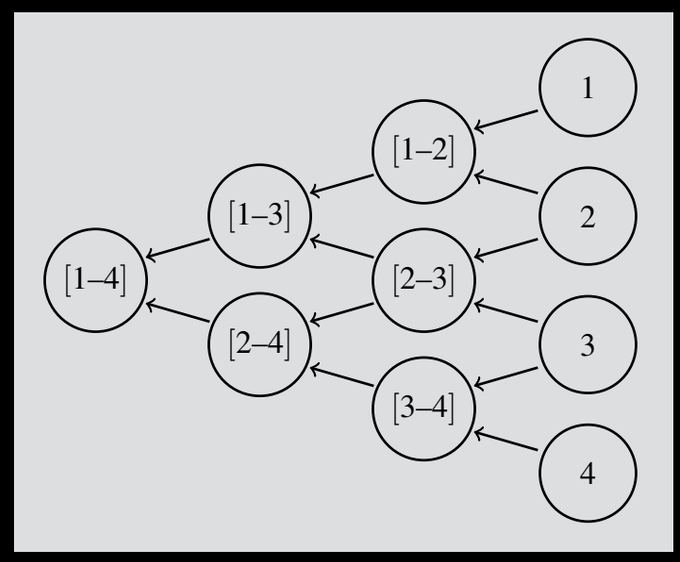
Complicated data



How to avoid discretisation?

These structures are similar

Can all be modelled as partial orders



Pattern mining

The goal

Find interesting and surprising patterns in massive data

The reality

Find massively many patterns in any data

Pattern mining

The goal

Find interesting and surprising patterns in massive data

The solution

Score patterns for their interestingness

Interestingness

What makes a pattern interesting ?

Very many scores (e.g., Geng & Hamilton 2006)

Support, confidence, Piatetsky-Shapiro / lift, accuracy, cosine, WRAcc, …

Interestingness

What makes a pattern interesting ?

Very many scores (e.g., Geng & Hamilton 2006)

Support, confidence, Piatetsky-Shapiro / lift, accuracy, cosine, WRAcc, chi-square, …

≈ unexpected (frequency of) co-occurrence

Interestingness

What makes a pattern interesting ?

≈ unexpected (frequency of) co-occurrence

Some ‘recent’ encompassing frameworks:

Information theory (see De Bie 2011, 2013)

Hypothesis testing (see Lijffijt et al. 2014)

Interestingness

What makes a pattern interesting ?

≈ unexpected (frequency of) co-occurrence

Some ‘recent’ encompassing frameworks:

Information theory (this paper)

Hypothesis testing

Interestingness

Information-theoretic interestingness

What makes a pattern interesting ?

≈ unexpected (frequency of) co-occurrence

= *Surprisal* = self-information = –log Pr(pattern)

Information-theoretic interestingness

What makes a pattern interesting ?

≈ unexpected (frequency of) co-occurrence

= *Surprisal* = self-information = –log Pr(pattern)

Actually

self-information / description length

One minor problem

Unexpected as compared to …?

Subjective interestingness

Users have different prior beliefs about data/domain

Data mining is an iterative process, users learn as analysis progresses

Information-theoretic framework (FORSIED) to deal with this introduced by De Bie (KDD 2011, IDA 2013)

General setting of our research project (ERC Grant FORSIED): Formalising Subjective Interestingness in Exploratory Data Mining [we are hiring a PhD student]

Subjective interestingness

Unexpected as compared to …

Maximum entropy distribution that satisfies the prior beliefs

Subjective interestingness

Unexpected as compared to …

Maximum entropy distribution that satisfies the prior beliefs

Prior beliefs in the form of expectations

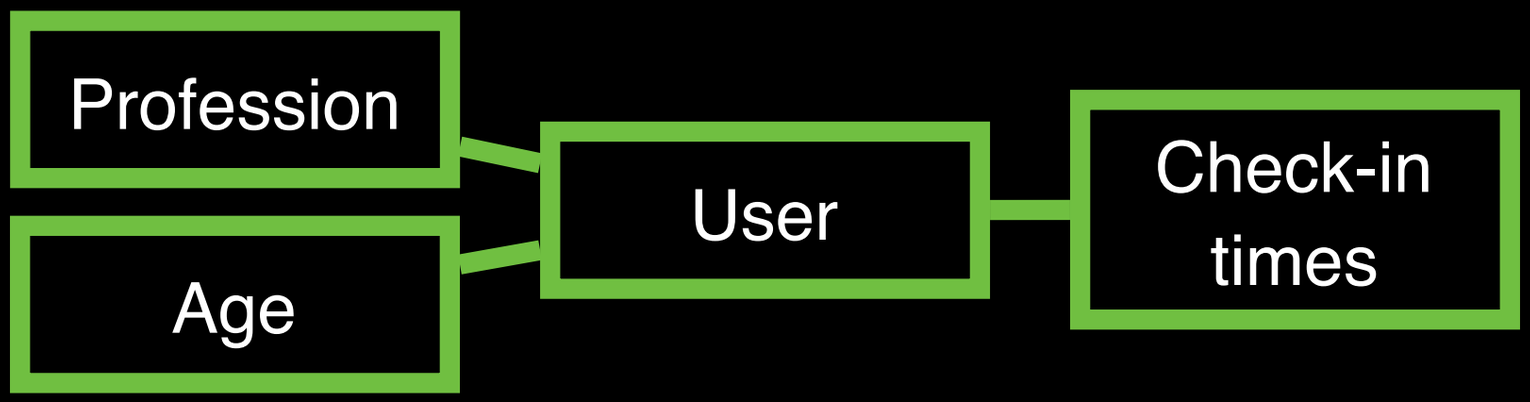
Row/column marginals / degree of nodes

First and second order moments for numerical vars

The following edges are present: …

Mining patterns

Relational Pattern Mining: RMiner (Spyropoulou et al. 2014)



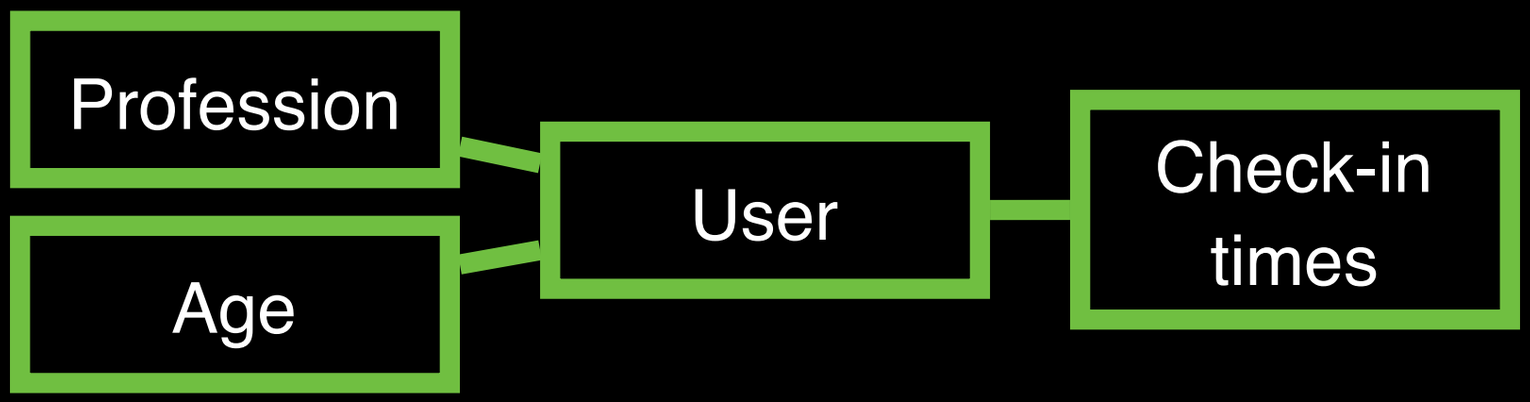
Relational: patterns across data tables with arbitrarily many links (many-to-many etc.)

Database = **entities** + **relationship instances**

Every user, age, profession, time is an *entity*

There is a *relationship instance* between a user and an age if that user has that age

Relational Pattern Mining: RMiner (Spyropoulou et al. 2014)



Relational: patterns across data tables with arbitrarily many links (many-to-many etc.)

Database = **entities** + **relationship instances**

Every user, age, profession, time is an *entity*

There is a *relationship instance* between a user and an age if that user has that age

Essentially a graph: nodes = entities, edges = relationship instances

Relational Pattern Mining: RMiner (Spyropoulou et al. 2014)

Database = entities + relationship instances

RMiner

Find all potentially interesting patterns

= all completely connected sets of entities

Rank the patterns for interestingness

Relational Pattern Mining: RMiner (Spyropoulou et al. 2014)

Database = entities + relationship instances

P-N-RMiner

Find all potentially interesting patterns

= all completely connected sets of entities

Rank the patterns for interestingness

Outline of algorithm

Instantiation of *fix-point enumeration* (Boley et al. 2010)

Works for any strongly accessible set system

All feasible generalisations of a set can be constructed by adding one element at a time

For every set except the empty set, an element exists that if removed, a feasible set is obtained

Proof for this in the paper

Closure operator is specific and dets. efficiency (optimal here?)

Ranking

Information-theoretic interestingness

What makes a pattern interesting ?

≈ unexpected (frequency of) co-occurrence

= *Surprisal* = self-information = –log Pr(pattern)

Actually

self-information / description length

Interestingness

For experiment, we used marginals as prior beliefs

The user knows the ‘frequency’ of entities, but not of the relations between entities

Background distribution is the maximum entropy distribution given the prior beliefs as constraints

Interestingness

It is straightforward to include other knowledge

Users may want to input their own ‘beliefs’

This is why interestingness is subjective

(Beliefs need not even be correct)

Or incorporate a pattern after reading it

Iterative data mining / pattern set mining

Case studies

Foursquare check-ins

Cheng et al. (ICWSM 2011)

P1: 1.6% of the users checked in frequently between [6am–7am], as well as [10.20am–10.50am]

P4: 4.5% of the users checked in frequently between [1.10am–2.30am], [4.30pm–6.30pm], as well as [8.30pm–9.30pm]

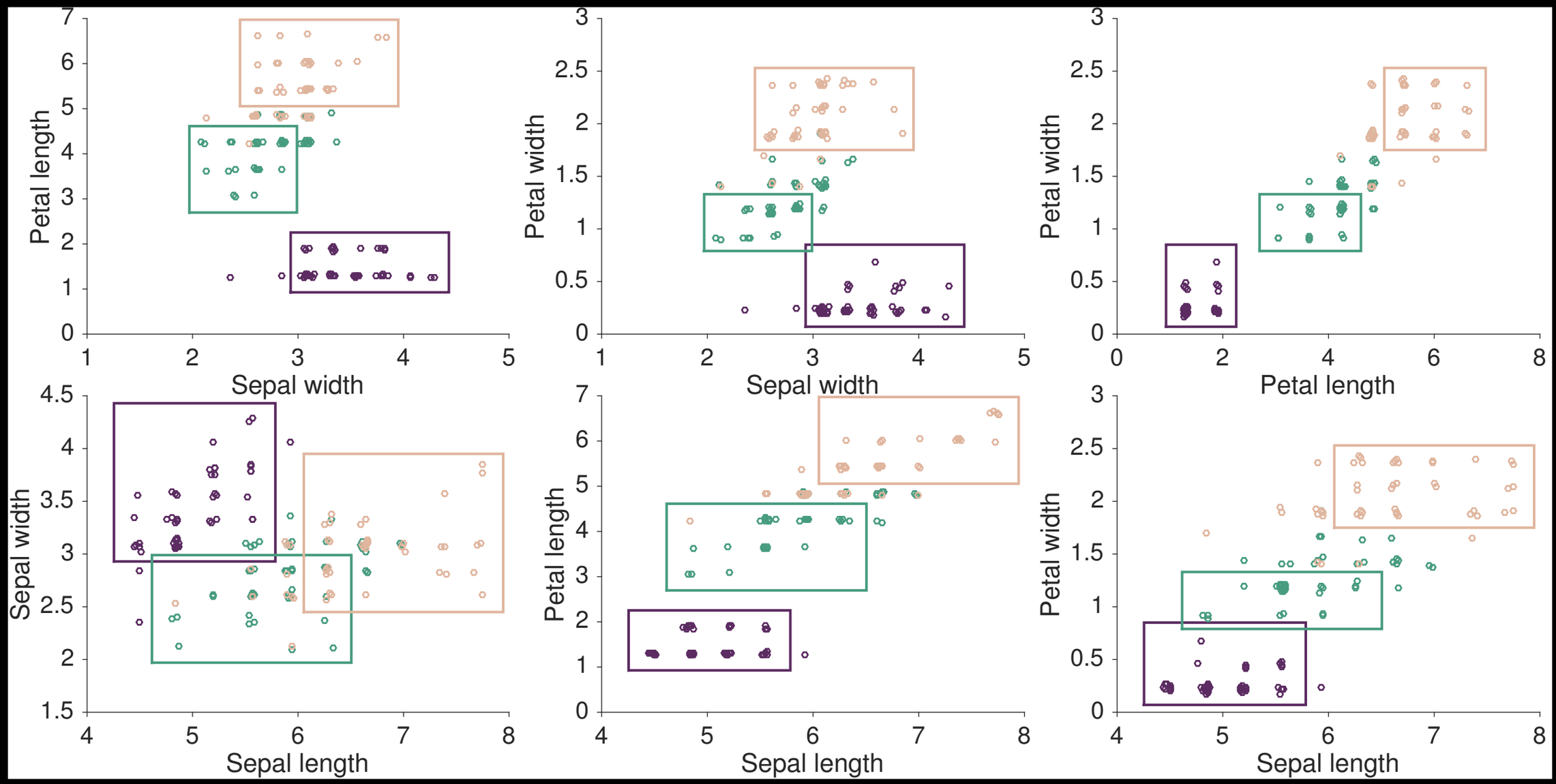
https://snap.stanford.edu/data/amazon-meta.html

Amazon book ratings

P1: 23 customers and 8 books, all of which are different versions of the book “Left Behind: A Novel of the Earth’s Last Days”, a rating [1–5] and the subjects *Fiction* and *Christianity*.

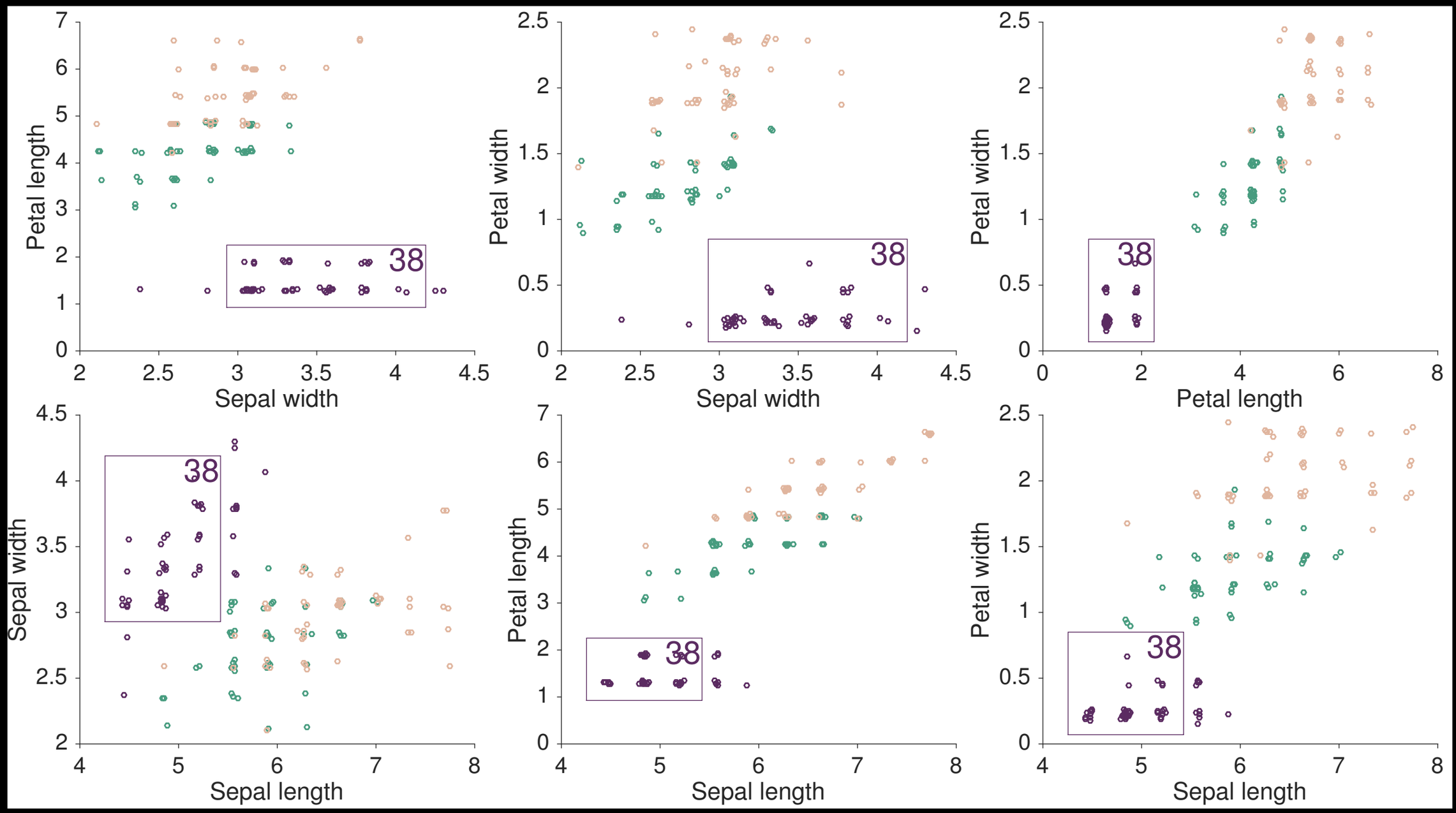
Amazon copies reviews between similar items

Subgroup discovery



How: take class label as mandatorily present

Subspace clustering



How: natural setting for P-N-RMiner

Summary

P-N-RMiner is a general solution for mining interesting patterns in data, supporting

Relational data

Structured attributes

Subjective prior beliefs

Iterative data mining

Thank you!

Scalability

There is polynomial-time delay between two closure steps (Spyropoulou et al. 2014)

We can capitalise on the partial order structures (this paper)

That actually gives a noticeable speed-up (this paper)

However, no polynomial-time delay between two outputs (proof will appear in follow-up)

This algorithm cannot be applied to very large data