Artificial
Intelligence &
Algorithms

Mining Constrained Regions of Interest: An optimization approach

Alexandre Dubray Guillaume Derval Siegfried Nijssen Pierre Schaus

Introduction

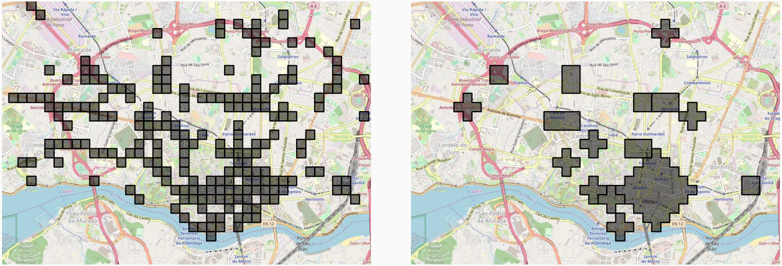
Motivations

- The amount of spatiotemporal data is exploding (smartphone applications, sports devices, fleet management, etc.)
- There is a need to process more efficiently these data
- Rewrite the raw trajectories (GPS points) as sequence of *Regions of Interest* (ROI)
- Multiple applications:
    - Trajectory pattern mining
    - Next location prediction
    - Urban management
    - ...

The general approach

1. Divide the map with a $N \times N$ grid.
2. Assign a density value to each cell. A cell is dense if its density is above a threshold. Multiple choice for the density:
    - Number of trajectory that passes in the cell
    - Number of trajectory that stays at least X minutes in the cell
    - etc.
3. Express the ROI as an aggregation of dense cells

Example of ROIs

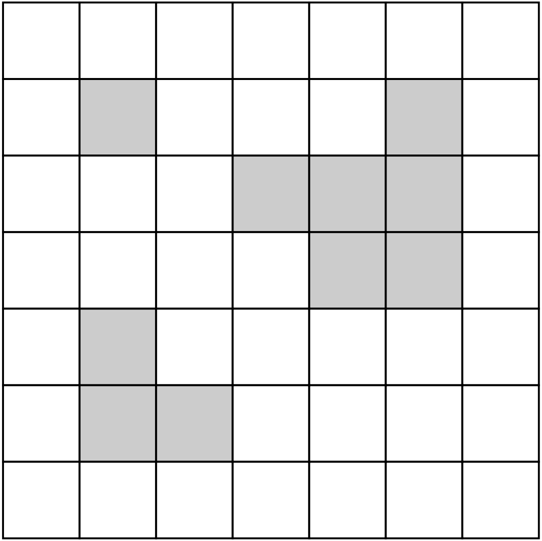


(a) Initial set of dense cells

(b) Solution found by our method

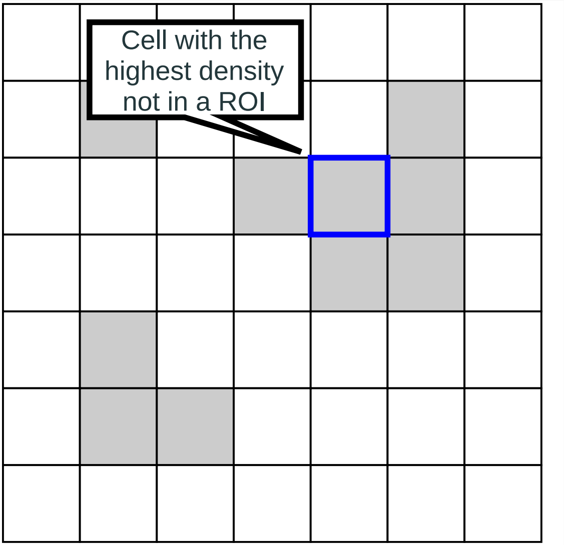
PopularRegion

Execution of the algorithm



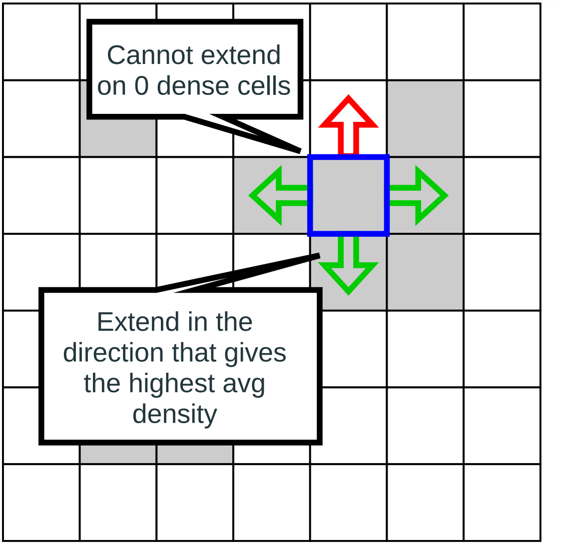
Fosca Giannotti et al. “Trajectory pattern mining”. In: *SIGKDD*. 2007

Execution of the algorithm



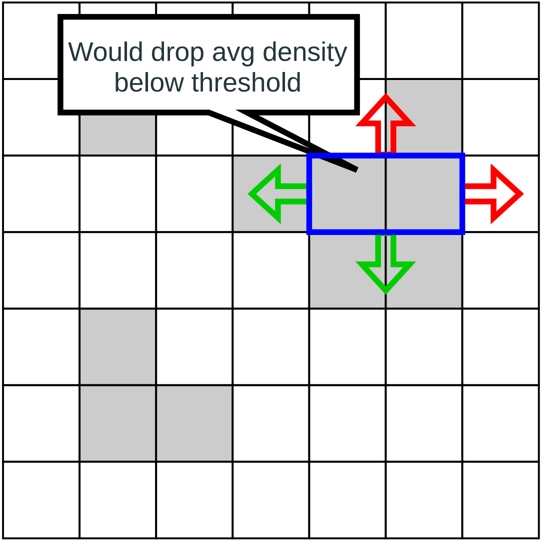
Fosca Giannotti et al. “Trajectory pattern mining”. In: *SIGKDD*. 2007

Execution of the algorithm



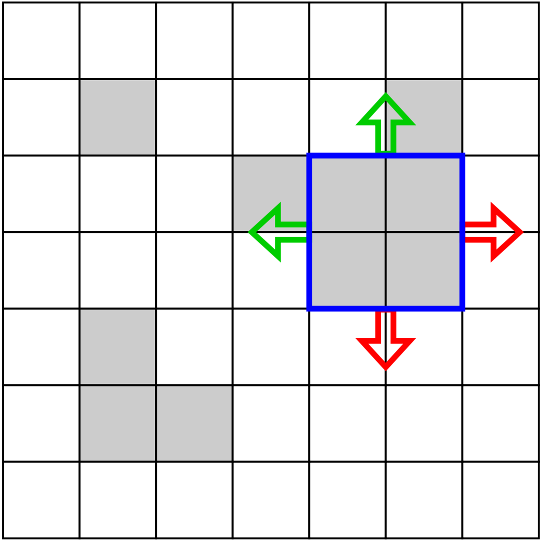
Fosca Giannotti et al. "Trajectory pattern mining". In: *SIGKDD*. 2007

Execution of the algorithm



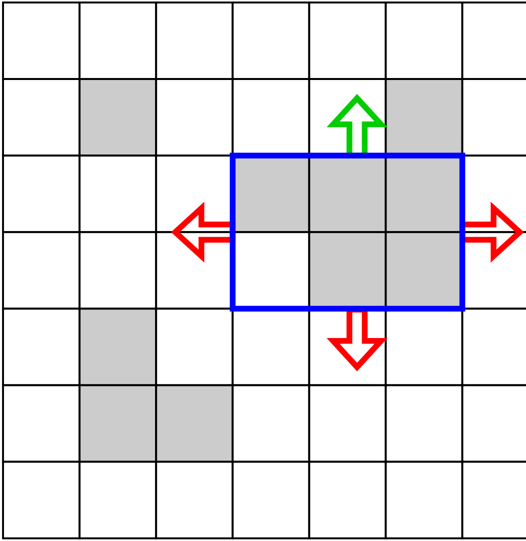
Fosca Giannotti et al. "Trajectory pattern mining". In: *SIGKDD*. 2007

Execution of the algorithm



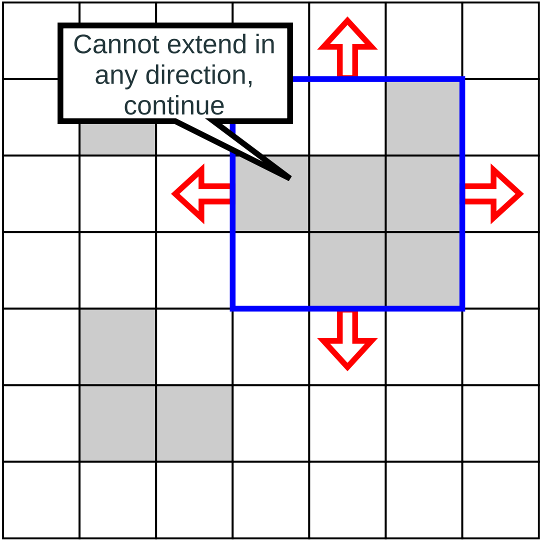
Fosca Giannotti et al. “Trajectory pattern mining”. In: *SIGKDD*. 2007

Execution of the algorithm



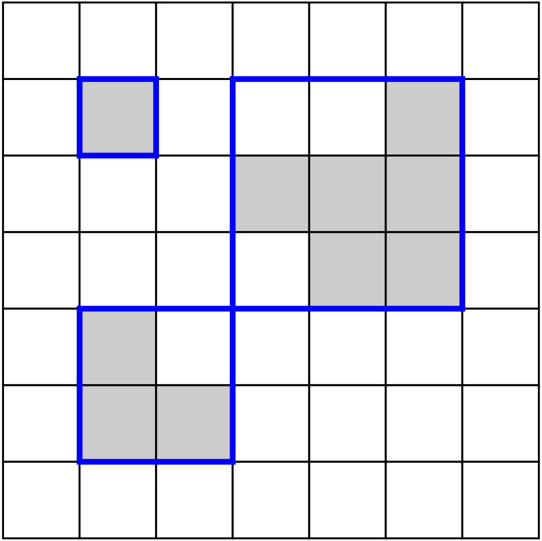
Fosca Giannotti et al. "Trajectory pattern mining". In: *SIGKDD*. 2007

Execution of the algorithm



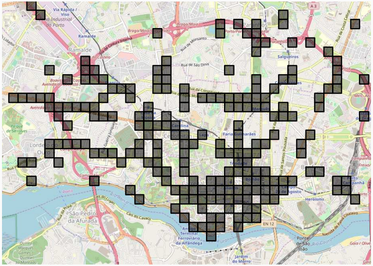
Fosca Giannotti et al. "Trajectory pattern mining". In: *SIGKDD*. 2007

Execution of the algorithm

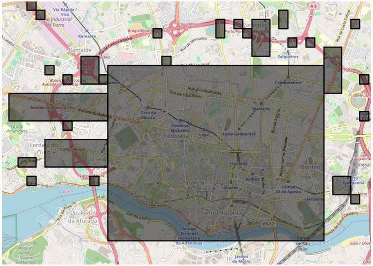


Fosca Giannotti et al. "Trajectory pattern mining". In: *SIGKDD*. 2007

Result of the algorithm



(a) Initial set of dense cells



(b) Solution with 5% min average density

Advantages and disadvantages

- Scalable
- Intuitive and good results for most configurations

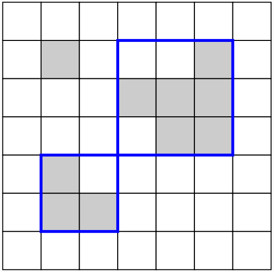
But...

- No formalization of the output
- Only rectangular regions
- Does not easily accept background knowledge
- Easy to create pathological input

Our method

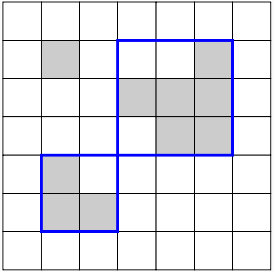
ROIs as an encoder

- The ROIs encode the dense status of the cells
- Example of encoding with two rectangles



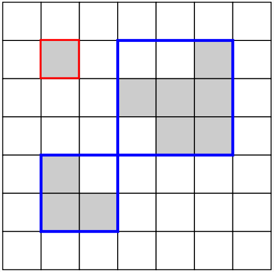
ROIs as an encoder

- The ROIs encode the dense status of the cells
- Example of encoding with two rectangles
- The encoding makes 5 errors



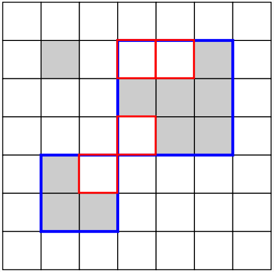
ROIs as an encoder

- The ROIs encode the dense status of the cells
- Example of encoding with two rectangles
- The encoding makes 5 errors
    - 1 dense cells is not covered



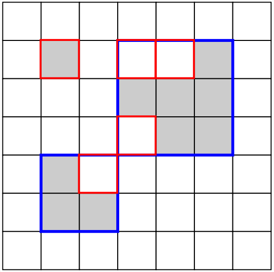
ROIs as an encoder

- The ROIs encode the dense status of the cells
- Example of encoding with two rectangles
- The encoding makes 5 errors
  - 1 dense cells is not covered
  - 4 non-dense cells are covered



ROIs as an encoder

- The ROIs encode the dense status of the cells
- Example of encoding with two rectangles
- The encoding makes 5 errors
  - 1 dense cells is not covered
  - 4 non-dense cells are covered
- We prefer encoding with fewer errors



Formalization of the problem (1)

Some notations:

- Let $\mathcal{G}$ be the grid, $\mathcal{G}^*$ the set of dense cells, $\mathcal{S}$ a set of candidates
- d_i (resp. u_i) is the number of dense (resp. non-dense) cells covered by the candidate $R_i \in \mathcal{S}$
- K is the maximum number of ROIs we want to find

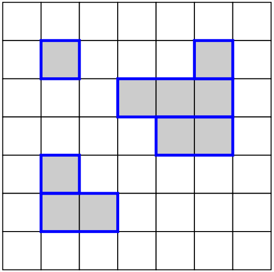
A first optimization model

$$
\begin{aligned}
\text{minimize} \quad & \overbrace{|\mathcal{G}^*| + \sum_{R_i \in \mathcal{S}} x_i \cdot \underbrace{(u_i - d_i)}_{\text{errors of the candidate}}}^{\text{errors of the model}} \\
\text{subject to} \quad & \\
\textstyle\sum_{R_i \in \mathcal{S}} x_i &\leq K \\
\textstyle\sum_{R_i \in \mathcal{S} | c \in R_i} x_i &\leq 1 \qquad \forall c \in \mathcal{G} \\
x_i &\in \{0, 1\} \quad \forall R_i \in \mathcal{S}
\end{aligned}
$$

Complexity of the models

We want to minimize the number of errors, but what about the complexity of the model?

- This model make no error but it requires 6 rectangles
- It does not represent well the dense cells
- We should limit the number of ROI to avoid these cases, but how to set K?



MDL Principle

- The Minimum Description Length (MDL) principle is a formalization of Ocam's razor
- The best hypothesis is the ones that compresses the most the data
- It is a two stages encoding:
  - Encode a model with length $L(M)$
  - Encode the data D given the model M with length $L(D \mid M)$
  - Best model is $\arg\min_M L(D \mid M) + L(M)$
- Trade-off between complexity of the model and generalization of the data

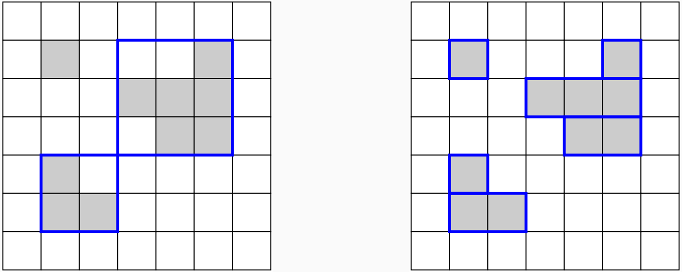
MDL for the ROIs

- Each cell is encoded with 2 integers (its row and its column)
- The length of a model, $L(M)$, is the sum of the length of the ROIs
  - A rectangle is encoded with two cells (4 integers)
  - A circle is encoded with one cell and a radius (3 integers)
  - Other forms have other encoding
- The length of data given a model, $L(D \mid M)$, is the number of errors it make
  - The ROIs tell which cells are dense
  - If we want to encode the grid, we need to encode the errors

MDL example

- $L(M) = 4 \cdot 2 = 8$
- $L(D \mid M) = 2 \cdot (4 + 1) = 10$
- $L(M) + L(D \mid M) = 18$

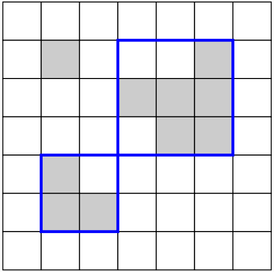
- $L(M) = 4 \cdot 6 = 24$
- $L(D \mid M) = 2 \cdot 0 = 0$
- $L(M) + L(D \mid M) = 24$



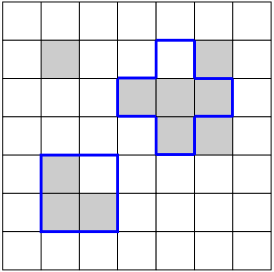
We prefer the model with 2 rectangles!

Example with circles

- $L(M) = 2 \cdot 4 = 8$
- $L(D \mid M) = 2 \cdot (4 + 1) = 10$
- $L(M) + L(D \mid M) = 18$



- $L(M) = 4 + 3 = 7$
- $L(D \mid M) = 2 \cdot (2 + 3) = 10$
- $L(M) + L(D \mid M) = 17$



We prefer the model with the circle

Formalization of the problem (2)

- Let $Sol \subseteq \mathcal{S}$ be a valid selection of candidates (the model) and $\mathcal{G}$ the grid (the data)
- $L(Sol) = \sum_{R_i \in Sol} size(R_i)$
- $L(\mathcal{G} \mid Sol) = 2(|\mathcal{G}^*| + \sum_{R_i \in Sol}(u_i - d_i))$
- Minimum Description Length principle says that the best solution is:

$$\arg\min_{Sol \in \mathcal{S}} L(\mathcal{G} \mid Sol) + L(Sol) = 2|\mathcal{G}^*| + \sum_{R_i \in Sol} (2(u_i - d_i) + size(R_i))$$

The final optimization model

$$
\begin{aligned}
&\text{minimize} \sum_{R_i \in \mathcal{S}} x_i \cdot \overbrace{(2(u_i - d_i) + size(R_i))}^{\text{Contribution to the description length}} \\
&\text{subject to} \\
&\textstyle\sum_{R_i \in \mathcal{S} | c \in R_i} x_i \leq 1 \qquad \forall c \in \mathcal{G} \\
&\qquad\qquad\quad x_i \in \{0, 1\} \quad \forall R_i \in \mathcal{S}
\end{aligned}
$$

The full method

1. Generate the set of candidates $\mathcal{S}$ (e.g. enumerate all distinct rectangle on the grid)
    - Candidate can have any shape
    - Compute their contribution to the description length
    - Apply *intra-ROI* constraints to filter the candidate set
2. Solve the optimization model
    - Model *inter-ROI* constraints with linear constraints in the ILP
    - Solve the ILP, the binary decision variables give the set of ROIs

Experiments

Setup

- Two versions of our method
  - With only rectangular regions
  - With rectangular and circular regions
- Showing results on Kaggle taxis dataset ($\approx$1.6 million trajectories)
- Comparing with PopularRegion[1] and OPTICS[2] (when clustering the dense cells)

[1] Fosca Giannotti et al. "Trajectory pattern mining". In: *SIGKDD*. 2007.
[2] Mihael Ankerst et al. "OPTICS: ordering points to identify the clustering structure". In: *ACM Sigmod record* (1999).

Execution time

Minimum density threshold	2%			5%		
Grid side size	100	150	200	100	150	200
Number of dense cells ($\lvert\mathcal{G}^*\rvert$)	571	597	537	230	178	137
Number of ILP candidates	23 814	7 779	3 399	2 880	1 232	434
ILP optimization time (s)	4.328	0.464	0.109	0.113	0.044	0.029
PopularRegion run time (s)	0.003	0.005	0.006	0.002	0.003	0.004
OPTICS run time (s)	0.209	0.222	0.200	0.084	0.065	0.051

Description Length

- For high density threshold, number of errors becomes similar
- ILP-based methods produce smaller models
- Overall the Description Length is inferior for ILP-based methods

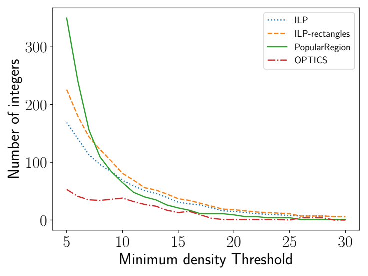


Figure 3: Encoding of the errors

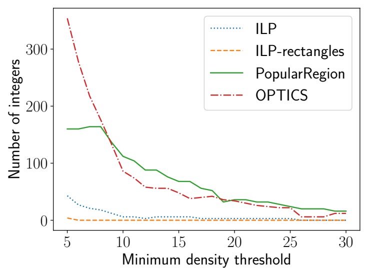


Figure 4: Encoding fo the models

Robustness to noise

- Start from a 100×100 grid
- Move every element of the trajectories to a new cell with a probability p
- Choose the new cell randomly in a square of size 10 around the initial cell
- Compute the solution from the noisy grid and compute the *precision*, *recall* and *F1-measure*.

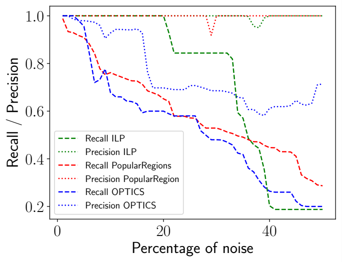


Figure 5: Recall and precision

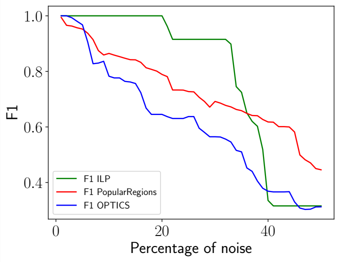


Figure 6: F1-measure

Conclusion and Future work

What we did:

- We propose an optimization model to find K ROIs from trajectory data
- Our method is more flexible than specific method since it accepts a wide range of constraints
- The runtime of the ILP becomes reasonable as long as there is not too much candidates
- Everything is Open Source, see `https://github.com/AlexandreDubray/mining-roi`

The next steps:

- Get rid of the grid
- Use the density information (instead of just dense/not dense)
- Provide support for more complex constraints