Scikit-mine: A pattern mining library in Python

Peggy Cellier

Scikit-mine
A pattern mining library in Python

UMR IRISA

Université de Rennes · INSA Rennes · cnrs · Inria

1. About the Scikit-mine project

01. Context

WEKA · KNIME · SPMF (An Open-Source Data Mining Library)

Pattern mining tools
(e.g., not exhaustive [2,3,4])

- Algorithms for discovering regularities
- Exploratory data science
- Unsupervised algorithms
- Tools not always easy to handle
- Mostly C/C++/Java implementations
- No popular libraries for easy combination with other Python libraries

scikit learn

Success story [1]

- Machine learning library
- Mostly supervised algorithms
- Python
- Huge popularity
- Supported by Inria

02. Project Scikit-mine

The Scikit-mine library [5]

- Focus on recent **MDL-based** pattern mining approaches [6]
  - Output few and meaningful patterns
  - Back up by Information Theory/statistics
- Python library **compatible with scikit-learn**
  - Easy to use for data scientists and practitioners
- Project started in January 2020
- 4 engineers worked on it

Scikit-mine — Pattern mining in Python

[1] https://scikit-learn.org/stable/
[2] https://www.cs.waikato.ac.nz/ml/weka/
[3] https://www.knime.com
[4] https://www.philippe-fournier-viger.com/spmf/
[5] https://github.com/scikit-mine/scikit-mine

2. How to use it

Scikit-mine is a Python module for pattern mining built on top of Pandas/Numpy/SciPy.

- Easy installation with pypi

```
$ pip install https://github.com/scikit-mine/scikit-mine.git
```

- Lot of tutorials (as Jupyter notebooks) and documentation online

3. What next?

Currently we are working on

- Adding more MDL-based algorithms (e.g., for graph datasets GraphMDL [10])
- Adding new Jupyter notebooks on medical/biology workflows (BidsProv project)

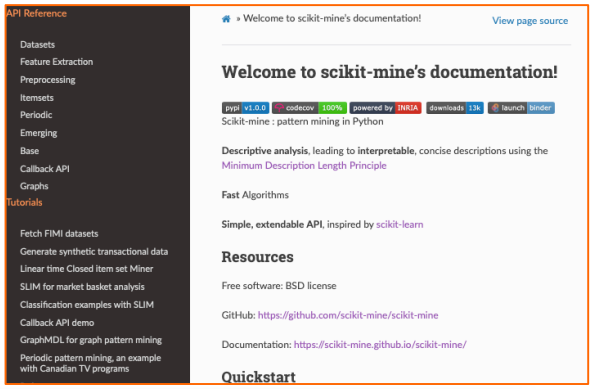
Further works

- Algorithms based on subjective interestingness [11]

4. Want to get involved?

Several ways to contribute

- **Tutorials**: documentation is key
  - Tell us your success stories in pattern mining
  - We will turn the most impactful/pedagogic ones in public notebooks
- **Development**
  - Implementation of algorithms
  - Mining speed will be important => Code reviews
- **Other ideas?**
  - Contact us!



Itemset mining

LCM [7]

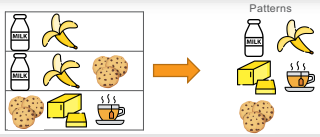
LCM is a classical itemset mining algorithm to efficiently extract **frequent closed itemsets**.

```
>>> from skmine.itemsets import LCM
>>> from skmine.datasets.fimi import fetch_chess
>>> chess = fetch_chess()
>>> lcm = LCM(min_supp=2000)
>>> patterns = lcm.fit_transform(chess)
>>> patterns.head()
    itemset  support
0      [58]     3195
1      [52]     3185
```

Itemset mining

SLIM [8]

SLIM is an **MDL-based algorithm** for the extraction of a **set of itemsets** that are well representative of the dataset.

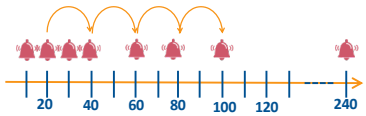



```
>>> from skmine.itemsets import SLIM
>>> D = [['bananas', 'milk'], ['milk', 'bananas', 'cookies'], ['cookies', 'butter', 'tea']]
>>> SLIM().fit(D).transform(D, singletons=True, return_tids=True)
          itemset    tids
0  [bananas, milk]  (0, 1)
1    [butter, tea]     (2)
2        [cookies]  (1, 2)
```

Sequence mining

Periodic Pattern Mining [9]

The Periodic Pattern Mining algorithm is an MDL-based algorithm for extracting a set of periodic cycles from event logs that characterises the periodic structure present in the data.




```
>>> from skmine.periodic import PeriodicPatternMiner
>>> import pandas as pd
>>> S = pd.Series("ring_a_bell", [10, 20, 32, 40, 60, 79, 100, 240])
>>> pcm = PeriodicPatternMiner().fit(S)
>>> pcm.transform(S)
   t0                    pattern  repetition_major  period_major  sum_E
0  20  (ring_a_bell)[r=5 p=20]                 5            20      2
```

Example

Classification examples with SLIM

- An integrated classifier in Scikit-mine is available and allows to solve binary and multiclass problems.
- It uses the SLIM compression algorithm.
- To use it, we need to have adiscretized dataset

In the example
- **Scikit-learn** to prepare the dataset
- **Scikit-mine** to learn a classifier
- **Scikit-learn** for the cross validation

```
from sklearn.model_selection import train_test_split

(X_train, X_test, y_train, y_test) = train_test_split(X, y, random_state=1, test_size=0.2, shuffle=True)
from skmine.itemsets.slim_classifier import SlimClassifier
items = set(item for transaction in X for item in transaction)
print("items", items)
clf = SlimClassifier(items=items)
clf.fit(X_train, y_train)
from sklearn.model_selection import cross_val_score

cross_validation = cross_val_score(clf, X, y, cv=10)
print(f"-> 10 Cross validation: {cross_validation.round(2)}")
print(f"-> Mean Accuracy : {round(cross_validation.mean()*100,1)} %")

-> 10 Cross validation: [0.93 0.93 0.87 0.93 0.93 0.93 1.   1.   1.   0.93]
-> Mean Accuracy : 94.7 %
```


Contact skm-dev@inria.fr

Contributors

CNRS
- Thomas Betton
- Hermann Courteille
- Cyril Regan

Inria
- Rémi Adon
- Luis Galárraga

University of Eastern Finland
- Esther Galbrun

University of Rennes 1
- Mensah-David Assigbi
- Francesco Bariatti
- Arnauld-Cyriaque Djedjemel
- Josie Signe
- Alexandre Termier

INSA Rennes
- Peggy Cellier (speaker)

Bibliography

- [6] Esther Galbrun. *The minimum description length principle for pattern mining: a survey*. Data Min. Knowl. Discov., 2022.
- [7] Takeaki Uno, Masashi Kiyomi and Hiroki Arimura (2004). *LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets*. Proc. IEEE ICDM Workshop on Frequent Itemset Mining Implementations, 2004.
- [8] Koen Smets and Jilles Vreeken. *Slim: Directly Mining Descriptive Patterns*. In Proc. of SIAM International Conference on Data Mining (SDM), 2012.
- [9] Esther Galbrun, Peggy Cellier, Nicolai Tatti, Alexandre Termier, Bruno Crémilleux. *Mining Periodic Patterns with a MDL Criterion*. In Proc. of ECML PKDD 2018.
- [10] Fransceco Bariatti, Peggy Cellier, Sébastien Ferré. *GraphMDL: Graph Pattern Selection based on Minimum Description Length*. In Proc. of IDA 2020.
- [11] Tijl De Bie. *Maximum entropy models and subjective interestingness: an application to tiles in binary databases*. Data Min. Knowl. Discov., 2011.

GitHub: https://github.com/scikit-mine/scikit-mine

Documentation: https://scikit-mine.github.io/scikit-mine/