FP-Growth Documentation

Release 1.0

Evan Dempsey

April 27, 2016

Contents

Contents:

CHAPTER 1

FP-Growth

A Python implementation of the Frequent Pattern Growth algorithm.

- Free software: ISC license
- Documentation: https://fp-growth.readthedocs.org.

1.1 Getting Started

You can install the package with pip:

```
pip install pyfpgrowth
```

Then, to use it in a project, inport it and use the find_frequent_patterns and generate_association_rules functions:

```
import pyfpgrowth
```

It is assumed that your transactions are a sequence of sequences representing items in baskets. The item IDs are integers:

```
transactions = [[1, 2, 5],
                [2, 4],
                [2, 3],
                [1, 2, 4],
                [1, 3],
                [2, 3],
                [1, 3],
                [1, 2, 3, 5],
                [1, 2, 3]]
```

Use find_frequent_patterns to find patterns in baskets that occur over the support threshold:

```
patterns = pyfpgrowth.find_frequent_patterns(transactions, 2)
```

Use generate_association_rules to find patterns that are associated with another with a certain minimum probability:

```
rules = pyfpgrowth.generate_association_rules(patterns, 0.7)
```

1.2 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

At the command line:

```
$ easy_install pyfpgrowth
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv pyfpgrowth
$ pip install pyfpgrowth
```

CHAPTER 3

Usage

To use FP-Growth in a project:

```
import pyfpgrowth
```

It is assumed that your transactions are a sequence of sequences representing items in baskets. The item IDs are integers:

```
transactions = [[1, 2, 5],
                [2, 4],
                [2, 3],
                [1, 2, 4],
                [1, 3],
                [2, 3],
                [1, 3],
                [1, 2, 3, 5],
                [1, 2, 3]]
```

Use find_frequent_patterns to find patterns in baskets that occur over the support threshold:

```
patterns = pyfpgrowth.find_frequent_patterns(transactions, 2)
```

Use generate_association_rules to find patterns that are associated with another with a certain minimum probability:

```
rules = pyfpgrowth.generate_association_rules(patterns, 0.7)
```

The FP-Growth algorithm uses a recursive implementation, so it is possible that if you feed a large transation set into find_frequent_patterns you will see a 'maximum recursion depth exceeded' error. If you do, you can modify your recursion limit:

```
import sys
sys.setrecursionlimit(some_value)
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/evandempsey/fp-growth/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

FP-Growth could always use more documentation, whether as part of the official FP-Growth docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/evandempsey/fp-growth/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *fp-growth* for local development.

1. Fork the *fp-growth* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/fp-growth.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv fp-growth
$ cd fp-growth/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 fp-growth tests
$ python setup.py test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/evandempsey/fp-growth/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_fp-growth
```

CHAPTER 5

Credits

5.1 Development Lead

- Evan Dempsey <me@evandempsey.io>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.1.0 (2016-04-25)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search